

UNIVERSIDADE FEDERAL DO PARANÁ

LUIZ HENRIQUE CARRILHO GONÇALVES

**PAGFIT**

**APLICATIVO DE PAGAMENTOS DIGITAIS**

CURITIBA

2018

LUIZ HENRIQUE CARRILHO GONÇALVES

**PAGFIT**  
**APLICATIVO DE PAGAMENTOS DIGITAIS**

Monografia apresentada como requisito parcial à obtenção do título de Especialista em Engenharia de Software no curso Pós-graduação em Especialização em Engenharia de Software, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientador: Prof. Dr<sup>a</sup>. Rafaela Mantovani Fontana

CURITIBA  
2018



MINISTÉRIO DA EDUCAÇÃO  
SETOR SEPT  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
CURSO DE PÓS-GRADUAÇÃO ENGENHARIA DE  
SOFTWARE

### TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE SOFTWARE da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **LUIZ HENRIQUE CARRILHO GONÇALVES** intitulada: **PAGFIT: CONSTRUINDO UM APLICATIVO DE PAGAMENTOS DIGITAIS**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 22 de Outubro de 2018.

RAFAELA MANTOVANI FONTANA  
Presidente da Banca Examinadora (UFPR)

JAIME WOJCIECHOWSKI  
Avaliador Interno (UFPR)

Dedico este trabalho à minha família, meus pais,  
à Deus e meus amigos que sempre incentivaram  
meu crescimento pessoal e apoiaram de  
forma ímpar nesta jornada.



## **AGRADECIMENTOS**

Primeiramente, agradeço a Deus por conceder as condições necessárias para a realização deste trabalho.

Agradeço à professora orientadora Dra. Rafaela Mantovani Fontana pela paciência, disposição e apoio durante todo o processo de desenvolvimento deste projeto, dispondo-se a orientar sempre, muitas vezes em contraturno.

Ao corpo docente do curso de Especialização em Engenharia de Software, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, especialmente ao coordenador professor Dr. Jaime Wojciechowski.

Por fim, agradeço à Universidade Federal do Paraná pela oportunidade de receber conhecimento através deste curso.

“No que diz respeito ao empenho, ao compromisso,  
ao esforço, à dedicação, não existe meio termo.  
Ou você faz uma coisa bem-feita ou não faz.”

Ayrton Senna

## RESUMO

Analisando-se a oportunidade de facilitar a consolidação de transações digitais foi desenvolvido o “PagFit”, concebido para ser utilizado como ferramenta de pagamento e recebimento por pessoas físicas por meio de dispositivos móveis. Em suas propostas estão a segurança de informações, performance, design agradável, compatibilidade em diferentes dispositivos e qualidade. Para isso, foi o React Native, tecnologia de desenvolvimento mobile que renderiza código Java nativo. O objetivo foi alcançado, levando em conta que o usuário pode transferir e receber dinheiro de forma simples, utilizando apenas seu dispositivo móvel, sem tarifas e sem burocracia.

**Palavras-chave:** pagamentos, transações, react native

## **ABSTRACT**

Identifying the opportunity to make digital transactions easier, "PagFit" app was developed. It is made to be a tool for payment and receipt through physical people using mobile devices. For this, the author has used React Native, a mobile development technology that renders into native Java code. The target was reached, considering the user is able to transfer and receive money in an easy way, using just his mobile device, tax-free and with no bureaucracy.

**Keywords:** payments, transactions, react native

## LISTA DE FIGURAS

Figura 1: Fluxo de uma transação.....	20
Figura 2: Tela de Login .....	30
Figura 3: Tela de cadastro .....	31
Figura 4: Dashboard.....	32
Figura 5: Tela de Extrato.....	33
Figura 6: Seleção de captura em dias.....	34
Figura 7: Gerar link de pagamento.....	35
Figura 8: Exibir QR code .....	36
Figura 9: Escanear código QR .....	37
Figura 10: Confirmação de pagamento .....	38
Figura 11: Pagamento concluído.....	39
Figura 12: Gerar boleto de recarga .....	40
Figura 13: Dados do boleto bancário .....	41
Figura 14: Agendar saque.....	42
Figura 15: Selecionar conta bancária para saque.....	43
Figura 16: Comprovante de solicitação de saque .....	44
Figura 17: Listar contas bancárias .....	45
Figura 18: Cadastrar conta bancária .....	46
Figura 19: Editar conta bancária .....	47
Figura 20: Diagrama de casos de uso negociais.....	52
Figura 21: Protótipo Login .....	56
Figura 22: Protótipo Dashboard .....	57
Figura 23: Protótipo da Tela de Extrato.....	58
Figura 24: Protótipo Gerar link de pagamento.....	59
Figura 25: Protótipo Código QR .....	60
Figura 26: Protótipo Realizar pagamento.....	61
Figura 27: Protótipo Autorizar pagamento.....	62
Figura 28: Protótipo Cadastrar conta .....	63
Figura 29: Diagrama de Classes Nível I.....	64
Figura 30: Diagrama de Classes Nível II.....	65
Figura 31: Diagrama de Classes Nível III.....	66
Figura 32: Diagrama de Sequência UC01.....	79

Figura 33: Diagrama de Sequência UC02.....	80
Figura 34: Diagrama de Sequência UC03.....	81
Figura 35: Diagrama de Sequência UC04 – Listar contas .....	82
Figura 36: Diagrama de Sequência UC05.....	83
Figura 37: Diagrama de Sequência UC06.....	84

## LISTA DE TABELAS

Tabela 1: Plano de Entregas .....	24
Tabela 2: Plano de Riscos .....	25
Tabela 3: Fluxo Git .....	28

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>9</b>
<b>LISTA DE TABELAS .....</b>	<b>11</b>
<b>1 INTRODUÇÃO .....</b>	<b>14</b>
1.1 PROBLEMA .....	15
1.2 OBJETIVO GERAL .....	16
1.3 OBJETIVOS ESPECÍFICOS .....	16
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>17</b>
2.1 HISTÓRIA DOS PAGAMENTOS .....	17
2.2 PROCESSO DE PAGAMENTO DIGITAL .....	18
2.3 QUICK RESPONSE CODE .....	21
2.4 SOLUÇÕES SEMELHANTES NO MERCADO .....	21
2.4.1 BANCO INTER .....	21
2.4.2 NUCONTA .....	22
2.4.3 OPA! PAGAMENTOS .....	22
2.4.4 MERCADO PAGO .....	22
<b>3 MATERIAIS E METODOS .....</b>	<b>23</b>
3.1 PROCESSO DE ENGENHARIA DE SOFTWARE .....	23
3.2 PLANO DE ATIVIDADE .....	24
3.3 PLANO DE RISCOS .....	24
3.4 MATERIAIS .....	26
3.4.1 DESENVOLVIMENTO .....	26
3.4.2 LINGUAGEM DE PROGRAMAÇÃO / SERVIDOR .....	26
3.4.3 TECNOLOGIAS RELACIONADAS .....	26
3.4.4 GERENCIAMENTO DE VERSÕES .....	26
3.4.5 ANÁLISE .....	28
3.5 DOCUMENTAÇÃO DO SISTEMA .....	28
<b>4 APRESENTAÇÃO DO SISTEMA .....</b>	<b>29</b>
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>48</b>
<b>REFERÊNCIAS .....</b>	<b>49</b>
<b>APÊNDICES .....</b>	<b>51</b>
APÊNDICE A: VISÃO INICIAL DO PROJETO .....	51
APÊNDICE B: DIAGRAMA DE CASOS DE USO NEGOCIAIS .....	52



APÊNDICE C: GLOSSÁRIO DO SISTEMA .....	54
APÊNDICE D: REGRAS DE NEGÓCIO .....	55
APÊNDICE E: PROTÓTIPO DAS PRINCIPAIS INTERFACES .....	56
APÊNDICE F: DIAGRAMA DE CLASSES DE OBJETOS DE NEGÓCIO .....	64
APÊNDICE G: DIAGRAMA DE CLASSES COM ATRIBUTOS .....	65
APÊNDICE H: DIAGRAMA DE CLASSES COMPLETO .....	66
APÊNDICE I: ESPECIFICAÇÃO DOS CASOS DE USO .....	67
APÊNDICE J: DIAGRAMAS DE SEQUÊNCIA.....	79
APÊNDICE K: MODELO FÍSICO DE DADOS.....	85
APÊNDICE L: PLANO DE TESTES .....	88
APÊNDICE M: CASOS DE TESTE .....	90
APÊNDICE N: LOG DE TESTES .....	91
APÊNDICE O: PLANO DE IMPLANTAÇÃO.....	93

## 1 INTRODUÇÃO

O avanço iminente das tecnologias de pagamentos e transferências digitais tem se acentuado na medida em que os sistemas operacionais e dispositivos móveis vão se aprimorando e oferecendo novas ferramentas. De acordo com uma pesquisa realizada pela Worldpay (2014, p. 13), os pagamentos utilizando carteiras digitais representaram até o início do ano de 2018 cerca de 12% do mercado, sendo adotado como um dos meios de pagamento de maior crescimento e adoção por parte dos consumidores.

Por meio de tecnologias de transações virtuais, têm surgido novas formas de transferir valores por meios digitais, como as transferências bancárias por meio de *internet banking*<sup>1</sup>, os links de pagamento digital, as transações digitais disponíveis em lojas virtuais e outras plataformas.

Nesse ínterim, algumas tecnologias foram surgindo, dentre eles o código *Quick Response*, também conhecido como *QR Code*, que será abordado com mais detalhes no decorrer deste trabalho, que consiste em uma imagem bidimensional com gravuras criptografadas que são reconhecidas pelos dispositivos através de leituras ópticas, como câmeras e leitores específicos.

Identificou-se, neste trabalho, a oportunidade de utilizar esta tecnologia para facilitar a transferência entre pessoas físicas, em que as partes escaneiam / geram este código e o sistema processa a transferência em poucos instantes.

Segundo o jornal Folha de São Paulo (2018), no ano de 2017 ao menos 62% dos brasileiros possuem um smartphone, o que representa, em dados atuais segundo o Departamento de Assuntos Econômicos e Sociais das Nações Unidas (REDAÇÃO, 2018), mais de 132 milhões de pessoas. Isso evidencia uma grande fatia de mercado consumidor, aliciando um grande fator para a proposta do sistema proposto.

---

<sup>1</sup> Portais digitais dos bancos que possibilitam ao usuário controlar suas contas por meio da Internet.

## 1.1 PROBLEMA

Em recentes pesquisas do IBGE (2017), entre 2015 e 2016, dos aproximadamente 102 milhões de brasileiros que possuíam um *smartphone*, apenas 57% utilizavam o dispositivo como principal ferramenta de acesso à Internet. Nos anos seguintes, de 2016 a 2017, os números aumentaram para 116 milhões e cerca de 77% dos brasileiros fazendo uso do *smartphone* como principal meio de acesso à rede.

Um artigo digital publicado pelo portal iMasters (2015) elucidou uma série de desvantagens de moedas físicas, tais como o desgaste das moedas, a facilidade de roubo e furto, dificuldades no controle e rastreamento e a possibilidade de falsificações. Frente a isso, as moedas digitais e as carteiras virtuais mostram-se mais seguras, simples e fáceis de usar.

Não localizando até o momento de concepção deste projeto nenhuma solução que fizesse uso de *QR code* como facilitador e autenticador de transações, o autor deste trabalho deu início ao projeto de *fintech*<sup>2</sup> PagFit.me. Nesse ínterim, identificou-se a oportunidade de conceber um modelo de negócio para tomar proveito desse levante para promover transações digitais seguras, sem tarifas e de forma livre.

Embora grandes bancos já tivessem soluções semelhantes, que serão apresentadas no capítulo seguinte, não havia aplicativo ou funcionalidade que permitisse ao usuário transacionar sem a necessidade de vincular-se a uma instituição financeira, comprovar renda, documentos, etc. Porém, para que o produto fosse validado, fez-se necessário a criação de um aplicativo prático, conciso e intuitivo que provesse todas as funcionalidades básicas.

Por mais que a tecnologia voltada à facilitação de transferências esteja evoluindo constantemente, ainda não há uma solução prática usando *smartphones* para transferir entre duas pessoas físicas. Existem soluções consolidadas há muito tempo, como a troca de dinheiro-vivo, as máquinas de cartão e mais recentemente, as maquininhas portáteis. Contudo, ainda se necessita de uma solução C2C - do inglês *customer to customer* - que seja prática e exclusiva.

---

<sup>2</sup> Do inglês *Financial Technologies*. Empresas de tecnologias que desenvolvem soluções voltadas ao mercado financeiro.

## 1.2 OBJETIVO GERAL

Desenvolver um aplicativo capaz de promover transações digitais entre pessoas físicas

## 1.3 OBJETIVOS ESPECÍFICOS

- Permitir que usuários possam criar um perfil gratuitamente no aplicativo
- Permitir aos usuários que mantenham contas bancárias para saques
- Permitir aos usuários criar links de pagamento por *QR code* ou código numérico
- Permitir aos usuários realizar pagamentos por *QR code* ou código numérico
- Permitir aos usuários manter o saldo da conta com recargas por boleto bancário ou saques para conta bancária

Este documento está organizado dentro da seguinte estrutura: no capítulo seguinte, o segundo, será apresentada a fundamentação teórica, definindo o conceito de pagamentos digitais e trazendo um panorama sobre o histórico e situação das tecnologias de transações digitais no mercado.

O terceiro capítulo é destinado à descrição do método, apresentando o processo de desenvolvimento do aplicativo em questão, os artefatos produzidos em cada etapa e quais as tecnologias que foram necessárias para a produção da solução apresentada.

No quarto capítulo serão apresentadas telas do aplicativo em produção, pronto para uso, bem como a explanação as funcionalidades entregues para os usuários.

Por fim, no quinto capítulo o autor conclui o trabalho, apresentando intenções para implementações futuras e considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos nos quais o projeto desenvolvido foi baseado. Explana-se a origem das transações físicas, da evolução dos sistemas financeiros e o surgimento das soluções digitais voltadas a pagamentos e recebimentos.

Ainda nesse sentido, será abordado o funcionamento da tecnologia *Quick Response Code*, que é chave para o aplicativo desenvolvido neste trabalho

### 2.1 HISTÓRIA DOS PAGAMENTOS

Desde que a sociedade foi tomando forma, fora necessária uma forma de controlar as trocas e o valor dos itens. Entre os séculos IX e X a.C, os fenícios começaram a fazer comércio exterior e o reinado passou a cunhar moedas, sendo depois adotado na Lídia Antiga pelo Rei Creso e pelos reinos gregos e persas. Desde então, de diversas maneiras foi-se evidenciando a necessidade de estabelecer meios seguros e lógicos para práticas de mercado.

Concomitantemente, a demanda pelo controle dessas transações comerciais foi tornando-se cada vez mais acentuada, forçando a criação de ocupações e instituições destinadas somente para essa finalidade, como os mercadores, trocadores de moedas e mais recentemente bancários, casas de câmbio, plataformas de pagamento e outros métodos de transações que estamos acostumamos a utilizar.

Em 1998, a empresa americana fabricante de softwares Confinity identificou a necessidade de realizar pagamentos por de cartão de crédito de forma muito segura por meio da Internet. Nesse íterim, o empresário sul-africano Elon Musk estava no comando da X.com, que promovia soluções bancárias digitais. No ano posterior, a Confinity lançou uma plataforma de pagamentos internos e dois anos depois, em 2000, a empresa de Musk comprou a mesma, dando origem a primeira companhia de transações digitais em larga-escala, a PayPal.

A empresa recém-formada surgiu como uma solução à crescente demanda de transferir valores através da Web sem a necessidade de um deslocamento físico. É neste patamar tecnológico que surgem as *fintechs*, empresas de tecnologia voltadas no desenvolvimento e consolidação de soluções para o mercado financeiro.

Como uma abstração da categoria de startups, tem sido adotado o termo *fintech*, que centraliza as empresas inovadoras que buscam oferecer soluções práticas para o mercado financeiro como um todo, seja voltado ao público final.

Deste ambiente nascem as grandes plataformas como a americana PayPal, a brasileira EBANX, a inglesa Stripe - dentre outras - que visam prover meios inteligentes e seguros para a consolidação de transferências digitais em âmbito internacional, com volumes grandiosos de valores e utilizados por dia.

## 2.2 PROCESSO DE PAGAMENTO DIGITAL

A fim de embasar o propósito deste trabalho, é importante definir o que são pagamentos digitais. A Unified Payment Interface for India - UPI trouxe uma definição bastante precisa:

*“Pagamentos digitais são uma forma de pagamento que são feitas através de métodos digitais de modo que tanto o pagador quanto o recebedor usam métodos digitais para enviar e receber dinheiro”. (UPI, 2017)*

Outra definição, proposta pela americana BTCA:

*“Pagamentos digitais são transferências de valores que são iniciadas e/ou recebidas utilizando dispositivos eletrônicos e canais para transmitir as instruções.” (BTCA, 2017)*

Transações digitais funcionam de uma forma simples e ocorrem de forma muito semelhante aos pagamentos físicos com cartão de crédito ou débito. Primeiramente, o pagador insere o cartão em um POS (*point of sale*), conhecidas popularmente como *maquininhas*, ou através de *checkouts*, que são geralmente o último passo de compras em lojas virtuais. Em seguida o adquirente, instituição que realiza o primeiro processo de consolidação da transação, valida os dados e submete para a bandeira do cartão do pagador. A bandeira por sua vez valida a autenticidade das informações recebidas e, utilizando critérios internos, autoriza a continuidade da transação. Na sequência, o último envia o processo ao banco emissor, que valida o perfil do pagador, verifica se há saldo ou limite suficiente e

adiciona mais uma camada de verificação antifraude, tal como conferência de perfil histórico e outros critérios semelhantes.

Caso todos os passos até o momento tenham sido bem-sucedidos, o adquirente é autorizado a receber o valor informado pelo POS. As tarifas operadas pela bandeira e pelo adquirente (e pelo banco em alguns casos) são repassados e o valor restante é depositado na conta bancária do recebedor ou em uma carteira virtual, dependendo do modelo de negócio do intermediador.

É importante frisar que nesse modelo, o recebedor pode ter o valor liquidado em prazos bastante extensos, definidos em contrato, como 21 dias, 30 dias ou em casos mais específicos, podem ultrapassar 90 dias. Essa modalidade é conhecida como D+X, onde D representa a data de consolidação da transação e X o número, em dias, para que o valor seja depositado na conta bancária do recebedor.

A Figura 1 apresenta o fluxo de uma transação digital:

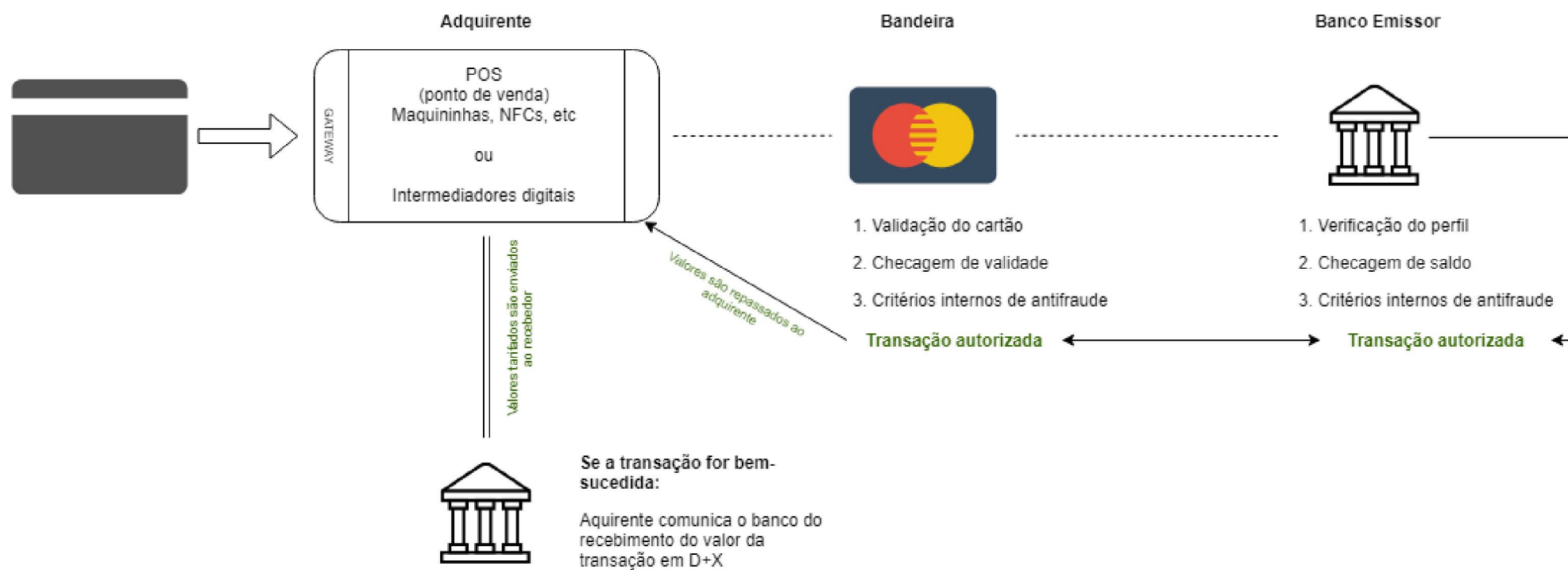


Figura 1: Fluxo de uma transação



## 2.3 QUICK RESPONSE CODE

Em 1994, a montadora japonesa Densowave identificou um gargalo no processo de catalogação dos componentes automotivos que eram utilizados em todo o processo industrial. Embora todos os produtos fossem devidamente registrados, tomava-se muito tempo para identificá-los no sistema.

No mesmo ano, a empresa lançou a primeira versão do *Quick Response Code*, que consiste em um padrão visual para que câmeras possam ler uma quantidade maior de dados de forma semelhante ao já consolidado código de barras.

O código consiste em um quadrado bidimensional onde em três dos quatro cantos, um quadrado menor é posicionado para que a câmera possa detectar o ângulo correto de escaneamento. Nas laterais desses quadrados menores, um pequeno código de barras que identifica qual versão do QR deve ser processada. Essa etapa é bastante importante, pois o leitor irá tomar conhecimento do tamanho da informação que está prestes a receber. Os símbolos restantes são a mensagem propriamente dita.

## 2.4 SOLUÇÕES SEMELHANTES NO MERCADO

Embora soluções que utilizem *QR code* como meio de comunicação para iniciar transações digitais, o aplicativo desenvolvido neste trabalho possui características que distinguem dos demais.

### 2.4.1 BANCO INTER

O aplicativo do Banco Inter possui uma funcionalidade, a Interpag, para transferir valores sem tarifas usando QR code. Porém a funcionalidade é acessível somente a correntistas do banco.

### 2.4.2 NUCONTA

O aplicativo do Nubank também possui uma funcionalidade para transferir valores sem tarifas usando *QR code*. Porém, da mesma forma, a funcionalidade é acessível somente a correntistas do banco.

### 2.4.3 OPA! PAGAMENTOS

O aplicativo do Opa! Pagamentos é o mais semelhante ao aplicativo deste projeto. Porém, tarifas são aplicadas a cada transação.

### 2.4.4 MERCADO PAGO

O aplicativo do Mercado Pago oferece a possibilidade de utilizar QR para recebimento, mas a funcionalidade está disponível para pessoas jurídicas e o código é relacionado ao usuário, ao invés do pagamento de fato.

No capítulo seguinte, são apresentados os materiais e métodos utilizados para o desenvolvimento do aplicativo proposto nesse projeto.

### 3 MATERIAIS E METODOS

A proposta da construção do Aplicativo PagFit tem como princípio que a facilitação de pagamentos abre portas para evolução financeira. Neste capítulo, aborda-se a metodologia adotada para o planejamento, projeto, desenvolvimento, testes e finalização do sistema.

#### 3.1 PROCESSO DE ENGENHARIA DE SOFTWARE

Propôs-se o desenvolvimento do sistema utilizando metodologias ágeis. A abordagem escolhida é o *Scrum*, metodologia ágil que defende o desenvolvimento iterativo do software, focando na entrega de funcionalidades mais necessárias de forma gradativa.

Embora o Scrum adote datas e/ou períodos específicos para reuniões, entregas e tomada de decisões para novas funcionalidades, o desenvolvimento do aplicativo em questão foi dado de maneira individual e, portanto, as funcionalidades foram sendo desenvolvidas à medida em que o autor definia quais eram as mais emergenciais e necessárias. A cada mescla no ramo principal no gerenciador de versões - que será detalhado nos próximos subcapítulos - eram definidas as próximas funcionalidades a serem desenvolvidas.

Seguindo os padrões da *Unified Modeling Language* (UML), o aplicativo foi concebido de forma a seguir as recomendações dos autores mais conhecidos, como GUEDES (2006). Concomitantemente ao desenvolvimento, os diagramas necessários na documentação foram sendo elaborados, a fim de possibilitar manutenção futura de forma mais eficaz.

Além disso, julgou-se importante o uso frequente de anotações e explicações nos mesmos, com o intuito de fazer conhecida a intenção dos desenvolvedores referente ao funcionamento de um determinado método ou a opção pela sequência lógica do caso de uso do sistema, entre outros exemplos.

### 3.2 PLANO DE ATIVIDADE

Inicialmente, foram decididas as tecnologias utilizadas para o desenvolvimento e *deploy* da aplicação, o escopo do projeto e deu-se início a prototipação de telas. Juntamente com o orientador, definiu-se um cronograma de *sprints* e quais seriam as funcionalidades entregues em cada período. A tabela a seguir reúne as entregas em cada *sprint*.

<b>Sprint</b>	<b>Objetivo principal ao final da Sprint</b>	<b>Data Início</b>	<b>Data Fim</b>
01	Ter todas as telas do aplicativo funcionais (não-operacionais)	07/05/2018	18/05/2018
02	Criar, editar, remover e autenticar na conta	21/05/2018	01/06/2018
03	Concretizar transação sem QR code	04/06/2018	15/06/2018
04	Gerar boleto de recarga de saldo e agendar saques para a conta	18/06/2018	29/06/2018
05	Gerar código personalizado de pagamento	02/07/2018	13/07/2018
06	Concretizar transação com QR code	16/07/2018	27/07/2018
07	Ambiente produtivo pronto e operacional, incluindo API	30/07/2018	10/08/2018
08	Compartilhar códigos em redes sociais e aplicativos	13/08/2018	24/08/2018
09	Visualizar histórico de transações	27/08/2018	07/09/2018
10	Obter melhor performance do aplicativo	10/09/2018	21/09/2018
11	Sprint destinada a correção de bugs/melhorias	24/09/2018	05/10/2018
12	Sprint destinada a correção de bugs/melhorias	08/10/2018	19/10/2018
Review	Aplicativo pronto	22/10/2018	26/10/2018

*Tabela 1: Plano de Entregas*

### 3.3 PLANO DE RISCOS

Entende-se por plano de riscos a abordagem e planejamento para tomada de decisões, considerando possíveis intempéries. Dessa forma, é possível mensurar o impacto dessas situações e qual a melhor forma de gerenciá-las e lidar com elas na iminência das mesmas.

Na tabela 2, estão listados os riscos e as abordagens relacionadas:

Risco	Consequência	Ação	Respons.	Probab.	Impacto	Classif.
Cronograma não realista	Inviabilidade de seguir o cronograma	Estimar o prazo de forma mais coerente e incluir folga para imprevistos	Autor	Média	Médio	5
Indisponibilidade e da equipe para reuniões	Atraso no cronograma	Verificar com antecedência a disponibilidade do autor e orientador;	Autor Orientador	Alta	Médio	6
Excesso na mudança de requisitos	Atraso no desenvolvimento do projeto	Especificar requisitos nas reuniões ou por meio das redes sociais de comunicação	Autor Orientador	Média	Alto	6
Dificuldade no uso das tecnologias	Atraso no desenvolvimento do projeto	Treinamento para aprendizagem; Escolha de tecnologias já treinadas	Autor	Alta	Alto	7
Abandono do projeto por um membro	Atraso no cronograma	Conversas preventivas e motivação em conjunto	Autor	Baixa	Alto	6
Atraso na entrega dos módulos	Atraso no cronograma	Refatorar cronograma com prioridades	Autor	Alta	Alto	7
Troca de tecnologias	Necessidade de refazer o código ou realizar adaptações	Optar sempre para tecnologias que ofereçam ganho de tempo.	Autor Orientador	Baixa	Alto	5
Falha na hospedagem do projeto	Atraso no cronograma	Realizar backups em diversas fontes.	Autor	Baixa	Alto	5

*Tabela 2: Plano de Riscos*

### 3.4 MATERIAIS

No subcapítulo a seguir estão descritos os materiais, tecnologias e tomadas de decisão que foram necessárias para dar início ao desenvolvimento do software em questão.

#### 3.4.1 DESENVOLVIMENTO

Após realizar-se uma pesquisa consistente das tecnologias adequadas para o desenvolvimento do software em questão, foi conveniente fazer o uso das seguintes ferramentas / tecnologias descritas a seguir.

#### 3.4.2 LINGUAGEM DE PROGRAMAÇÃO / SERVIDOR

Uma das restrições impostas pela coordenação do curso proponente é a da linguagem ser, impreterivelmente, *Java*. Portanto, optou-se por uma tecnologia capaz de produzir código consistente em *Java* integrando com uma interface de aplicação (*API*) externa disponível apenas para o autor e os interessados neste projeto. A tecnologia que melhor atendeu a este cenário foi o *React Native*

#### 3.4.3 TECNOLOGIAS RELACIONADAS

Uma vez que o sistema não irá integrar a um banco de dados, fazendo uso de uma *API* externa para providência de dados, o sistema faz uso do *Javascript* no *frontend* com o framework *React Native* e de pacotes livres como o *Axios*, *React Navigation* e outros pacotes *Node*.

#### 3.4.4 GERENCIAMENTO DE VERSÕES

A fim de desenvolver o sistema de forma eficaz e eficiente, foi proposto que, no fluxo de desenvolvimento, fosse utilizado um gerenciador de versões. Após

debates, julgou-se apropriado utilizar o *GitHub*, que utiliza a tecnologia *Git* para promover o gerenciamento das versões de código.

O *workflow* do repositório é dividido em duas *branches* principais e inúmeras secundárias.

Na *branch* mais estável, chamada *master*, é armazenada o produto mais estável do sistema.

A *branch develop* é constantemente atualizada pelas *branches* secundárias. Esta armazena a produção mais estável. Assim que houver uma versão produzida que possa ser utilizada, realiza-se o *merge* com a *master*.

As *branches* secundárias são divididas em quatro tipos: *feat*, *fix*, *spec* e *hotfix*.

- *Feat*: *branch* que carrega novas funcionalidades no sistema. Ex: *feat\_calcula\_saldo*
- *Fix*: *branch* que atualiza funcionalidades que apresentavam erros ou estavam incompletas. Ex: *fix\_timezone\_usuario*.
- *Hotfix\**: *branch* que carrega correções críticas que afetar a integridade do sistema. Ex: *hotfix\_login\_administrador*.
- *Spec*: *branch* que carrega uma estrutura necessária para criar funcionalidades. Ex: *spec\_dao\_contabancaria*.

\* Por se tratar de uma correção crítica de extrema urgência, a *hotfix* é a única *branch* que realiza *merge* direto na *master*.

A tabela 3 sintetiza o fluxo de desenvolvimento adotado no Git:

Nome	Descrição	Proteção	Recebe merge	Realiza merge
master	produto estável	protegida (todos)	develop, hotfix	--
develop	produção estável	protegida	secundárias	master
feat_*	novas funções	--	--	develop
fix_*	correções	--	--	develop
hotfix_*	crítica	--	--	develop, master
spec_*	estrutura	--	--	develop

*Tabela 3: Fluxo Git*

### 3.4.5 ANÁLISE

Para elaboração dos diagramas, foram utilizados dois softwares. Para os diagramas entidade-relacionamento e de modelo físico, foi utilizado o *Draw.io*.

## 3.5 DOCUMENTAÇÃO DO SISTEMA

Ao longo do projeto, foram desenvolvidos os diagramas e artefatos necessários para uma análise e especificação completa do sistema.

No início do projeto, definiram-se o documento de visão, que pode ser visualizado no “Apêndice A”, os casos de uso (Apêndice B), glossário (Apêndice C), regra de negócio (Apêndice D) e a prototipação de telas (Apêndice E).

Na sequência, foram desenvolvidos os diagramas de classes (Apêndice F, G e H) e foram especificados os casos de uso de forma mais detalhada (Apêndice I). Os diagramas de sequência também foram modelagem nessa etapa (Apêndice J).

Por fim, os testes e implementação do sistema foram realizados conforme os apêndices L, M, N e O.



## 4 APRESENTAÇÃO DO SISTEMA

O aplicativo PagFit é um sistema mobile desenvolvido em React Native, compilado em Java Android. Os requisitos mínimos para efetuar um deploy são:

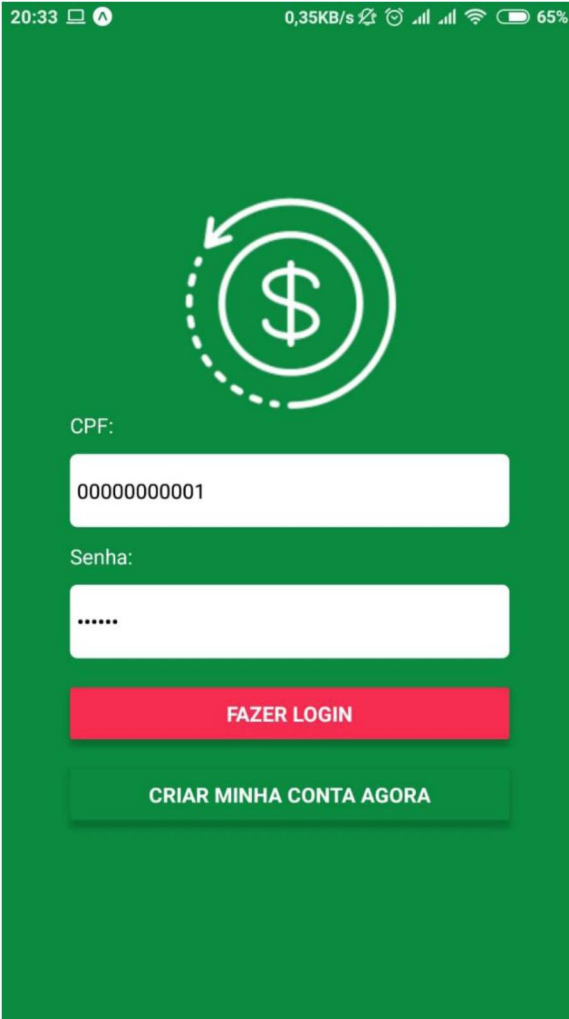
- Android Studio
- Java SDK e JVM
- Smartphone com Android 4 ou superior ou iPhone 10 ou superior
- Node.js 8 ou superior

O PagFit foi desenvolvido de modo a atender um tipo de usuário: pessoa física. Para facilitar a compreensão sobre as funcionalidades do aplicativo, estão relacionadas a seguir as interfaces finais referentes aos casos de uso especificados (disponíveis no apêndice deste trabalho).

Entende-se também que:

- Um usuário pode criar uma conta gratuitamente
- Um usuário pode receber e/ou efetuar pagamentos
- Um usuário pode emitir boletos de recarga de saldo
- Um usuário pode agendar saques para sua conta bancária

Dessa forma, o sistema foi construído para que siga a estrutura lógica acima citada.



20:33 0,35KB/s 65%

CPF:

00000000001

Senha:

.....

FAZER LOGIN

CRIAR MINHA CONTA AGORA

*Figura 2: Tela de Login*

Logo ao abrir o aplicativo, é apresentada a tela de login (Figura 2). O usuário pode efetuar o login ou criar uma conta nova.

20:33 0,05KB/s 65%

Nome:

Luiz Henrique Carrilho Gonçalves

Email:

dev.luizh@gmail.com

Número de celular:

41998495613

CPF:

12312312387

Senha:

.....

Senha de quatro dígitos:

....|

**CRIAR MINHA CONTA AGORA**

*Figura 3: Tela de cadastro*

Ao selecionar a opção de criar uma conta, o usuário é redirecionado para a tela de cadastro (Figura 3). Nela, são apresentados os campos cadastrais: nome, e-mail, número de celular, CPF, senha para login e senha de quatro dígitos, que será utilizado na validação de transações.



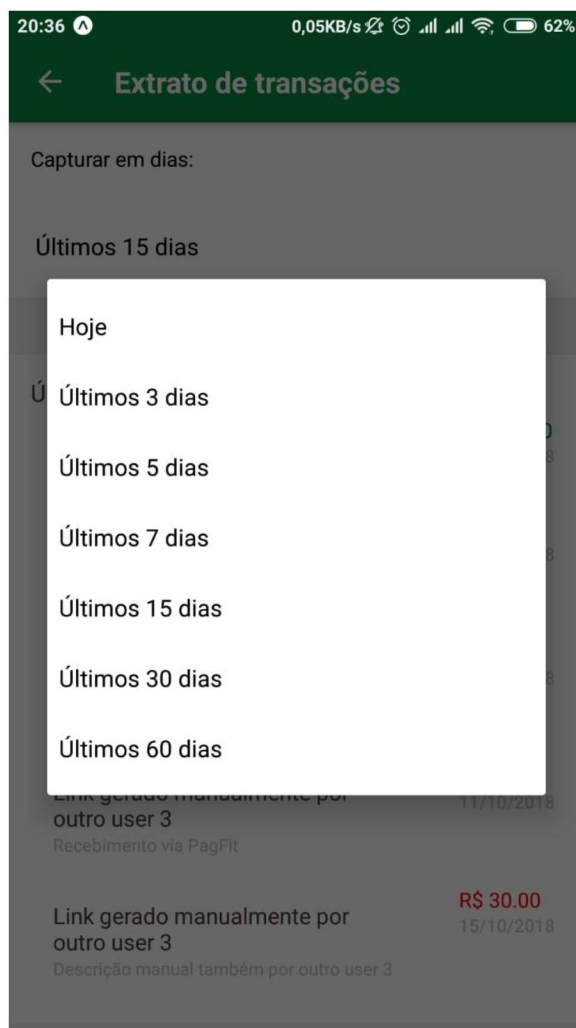
Figura 4: Dashboard

Ao realizar o login ou finalizar o cadastro, o usuário é direcionado a tela de Dashboard (Figura 4). Nela é apresentado o saldo e as transações dos últimos três dias. A partir dessa tela, é possível controlar menu lateral, exibido através do movimento “arrasta-e-solta” do lado esquerdo.



Figura 5: Tela de Extrato

O usuário pode consultar as transações realizadas através da tela de Extrato (Figura 5). São apresentadas todas as transações registradas no intervalo selecionado. O usuário pode alterar o intervalo de captura selecionando o campo “Capturar em dias” (Figura 6).



*Figura 6: Seleção de captura em dias*

Nenhuma transação é removida do sistema, porém a consulta no extrato é limitada a sessenta dias.

20:36 0,10KB/s 62%

← Gerar link de pagamento

Valor total:  
R\$15,00

Referência:  
Pizza

Limite de recebimento:  
Quantas pessoas podem realizar esse pagamento?:  
4

GERAR LINK

# \$ % \_ - 1 2 3 ?  
@ ( ) = + 4 5 6 !  
{&= ' : % / 7 8 9 < x  
abc " \_ \* , 0 . <

Figura 7: Gerar link de pagamento

O usuário, para poder receber dinheiro de outros usuários, deve criar links de pagamento. Para isso, ele pode acessar a tela “Gerar link de pagamento” (Figura 7). O usuário além de fornecer o valor e a referência, pode configurar o limite de recebimento para aquele link.

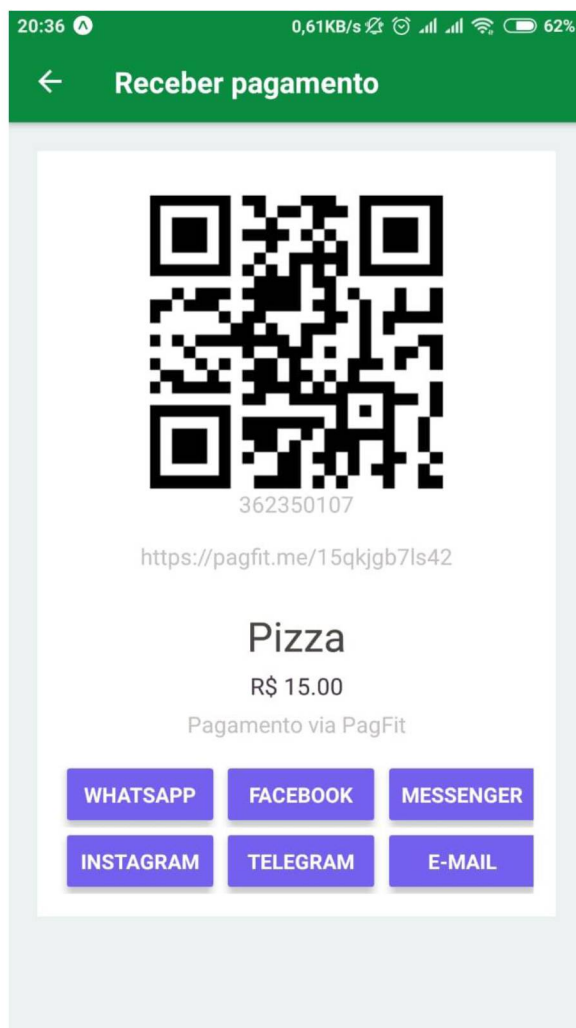
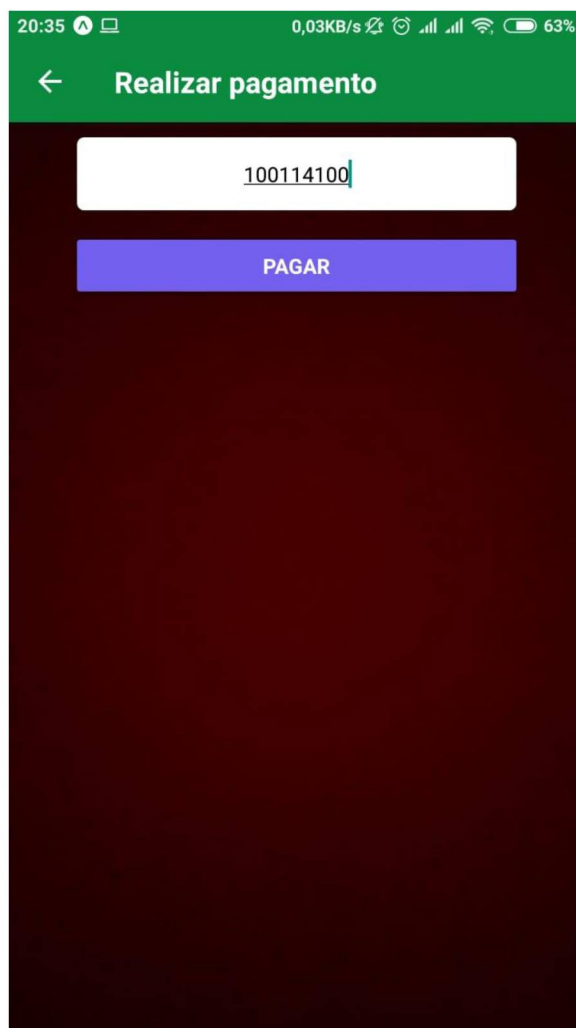


Figura 8: Exibir QR code

Após criar um link de pagamento, o usuário é direcionado para a tela de exibição do QR code (Figura 8). Além do código QR, é apresentado um PIN de 9 dígitos que pode ser utilizado para o pagamento, bem como o link que é acessível em navegadores web.

O usuário também pode enviar facilmente para outros aplicativos, por meio dos botões apresentados na parte inferior da tela.





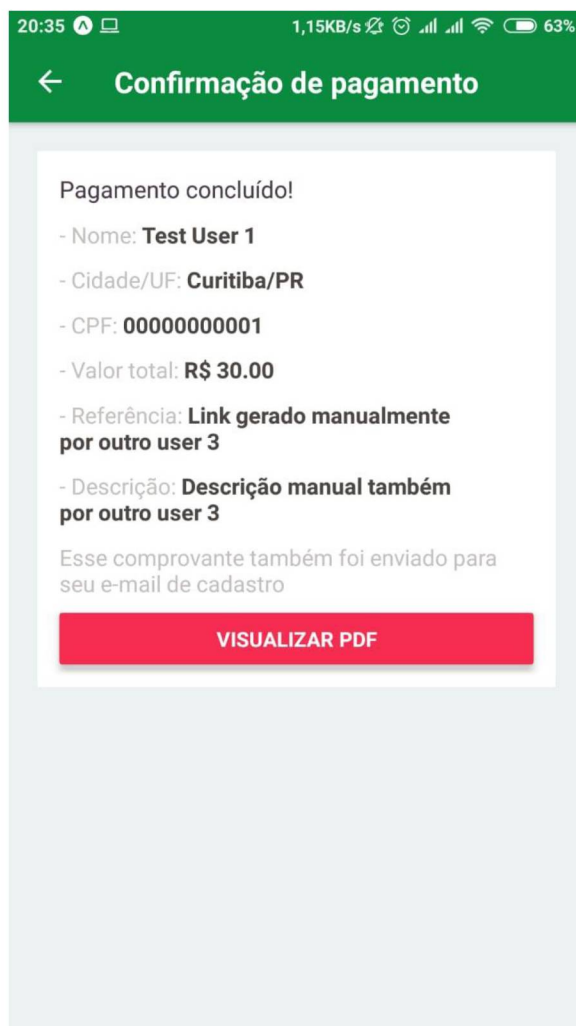
*Figura 9: Escanear código QR*

O usuário, para poder realizar um pagamento para outro usuário, deve inserir um código PIN ou escanear um código QR. Para isso, ele pode acessar a tela “Realizar pagamento” (Figura 7).



*Figura 10: Confirmação de pagamento*

Uma vez que o link é corretamente processado pelo sistema, o aplicativo redireciona para a tela de confirmação de pagamento (Figura 10). Os dados do link são apresentados na tela e usuário é orientado a inserir uma senha de quatro dígitos numéricos.



*Figura 11: Pagamento concluído*

Ao finalizar o pagamento, o sistema apresenta as informações de confirmação de pagamento (Figura 11). O usuário pode realizar o download do comprovante em PDF.

20:37 0,08KB/s 62%

← Recarga de conta

Valor total:

R\$50,00

Tarifas administrativas poderão ser acrescentadas ao valor;

GERAR BOLETO

# \$ % \_ - 1 2 3 ?

@ ( ) = + 4 5 6 !

{&= ' : % / 7 8 9 < x

abc " \_ \* , 0 . <

Figura 12: Gerar boleto de recarga

Para recarregar o saldo de sua conta, o usuário deve gerar um boleto bancário. Para isso, ele deve acessar a tela de “Recarga de conta” (Figura 12) e inserir o valor de recarga desejado.



*Figura 13: Dados do boleto bancário*

Ao preencher o formulário anterior, o aplicativo apresenta a linha digitável do boleto bancário (Figura 13). O usuário pode optar por fazer o download em PDF.

20:37 1,03KB/s 62%

### ← Saque para conta bancária

Valor total:

R\$50,00

Lembre-se que essa é uma solicitação e deverá ser aprovada pelo nosso time

Conta bancária:

Banco Bradesco S.A. 588 545066

**SOLICITAR SAQUE**

# \$ & \_ - 1 2 3 ?

@ ( ) = + 4 5 6 !

{&= ' : % / 7 8 9 < x

abc " \_ \* , 0 . <

Figura 14: Agendar saque

Para retirar o saldo de sua conta, o usuário deve solicitar um saque para uma conta-corrente bancária. Para isso, ele deve acessar a tela de “Solicitar saque para conta bancária” (Figura 14) e inserir o valor de saque desejado. O usuário também deve indicar qual a conta para saque.

20:37 0,01KB/s 62%

← Saque para conta bancária

Valor total:

R\$50,00

Lembre-se que essa é uma solicitação e deverá ser aprovada pelo nosso time

Conta bancária:

Banco Bradesco S.A. 588 545066

SOLICITAR SAQUE

Banco Bradesco S.A. 588 545066

# \$ & \_ - 1 2 3 ?

@ ( ) = + 4 5 6 !

{&= ' : % / 7 8 9 < x

abc " \_ \* , 0 . <

Figura 15: Selecionar conta bancária para saque

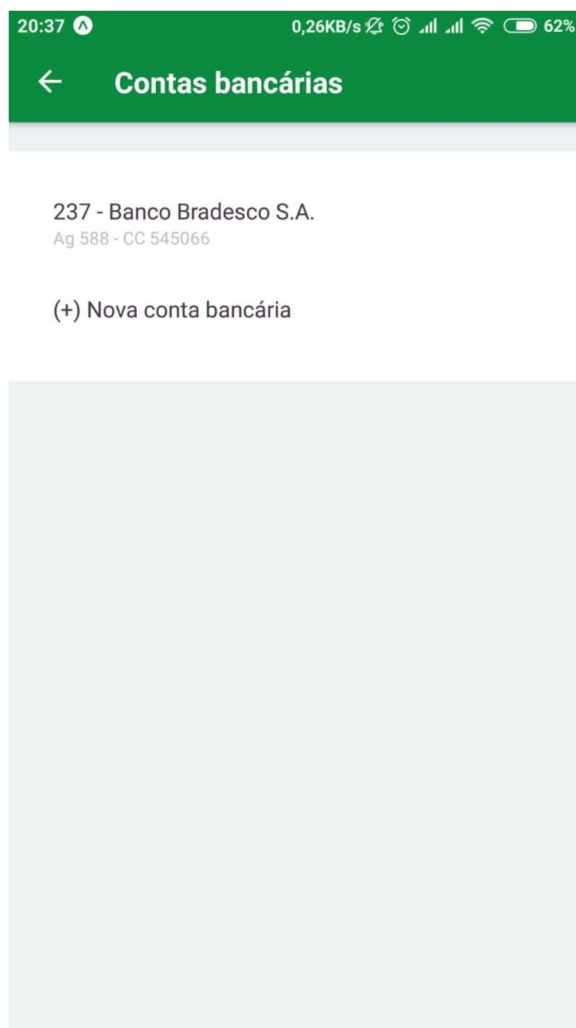
O usuário pode selecionar, dentre as contas bancárias cadastradas, qual deseja que o saque seja realizado (Figura 15).



*Figura 16: Comprovante de solicitação de saque*

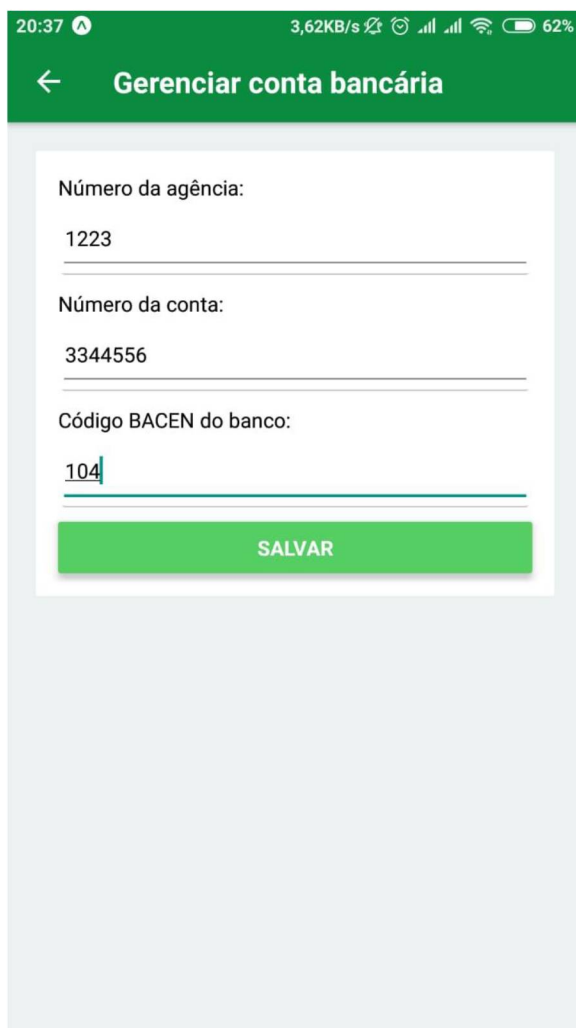
Ao preencher o formulário anterior, o aplicativo apresenta o comprovante de solicitação de saque para conta bancária (Figura 16). O usuário pode optar por fazer o download em PDF.





*Figura 17: Listar contas bancárias*

O usuário pode consultar as contas bancárias que ele administra (Figura 17). Ao acessar a tela, são apresentados o número da conta, agência, banco e código Bacen do banco de cada uma das contas.



20:37 3,62KB/s 62%

← Gerenciar conta bancária

Número da agência:

1223

Número da conta:

3344556

Código BACEN do banco:

104

SALVAR

*Figura 18: Cadastrar conta bancária*

Ao tocar no botão “(+) Nova conta bancária”, é apresentado o formulário de cadastro de conta bancária. O usuário deve informar número da agência, número da conta e número do banco.

The image shows a mobile application screen with a green header bar. The header contains a back arrow icon on the left and the text "Gerenciar conta bancária" in the center. Below the header, there is a white form area. The form contains three input fields, each with a label and a value: "Número da agência:" with the value "1223", "Número da conta:" with the value "3344556", and "Código BACEN do banco:" with the value "104". Below the input fields, there are two buttons: a green button labeled "SALVAR" and a red button labeled "EXCLUIR". The status bar at the top of the screen shows the time "20:38", a signal strength indicator, a data speed indicator "0,73KB/s", a battery level indicator "62%", and a battery icon.

20:38 0,73KB/s 62%

← Gerenciar conta bancária

Número da agência:  
1223

Número da conta:  
3344556

Código BACEN do banco:  
104

SALVAR

EXCLUIR

*Figura 19: Editar conta bancária*

O usuário também pode manter os dados de uma conta bancária, pressionando sobre a linha correspondente à conta na listagem (Figura 17). Em seguida, será apresentada a tela de edição (Figura 19).

## 5 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi desenvolver um aplicativo mobile com interface simples, para promover pagamentos através de QR code.

Para atender a esse objetivo, fez-se necessário a identificação dos problemas relacionados aos processos de pagamento e uma pesquisa bibliográfica, constatando-se que a tecnologia pode proporcionar melhoria na segurança e controle de transações entre pessoas físicas.

Para este projeto foi utilizado como metodologia de desenvolvimento o Scrum, seguindo as suas diretrizes como levantamento de requisitos, análise de requisitos, projeto do Sistema, programação do sistema e testes do sistema.

Ainda há muito o que crescer neste projeto. Dentre as implementações possíveis estão a integração com cartões de crédito, modelos de assinaturas, suporte a contas de pessoas jurídicas e uma interface web.

Observou-se que os objetivos propostos para este trabalho foram atingidos, agregando conhecimentos aprofundados na metodologia de desenvolvimento de software e agregou diversos conhecimentos técnicos.

## REFERÊNCIAS

BAGNALL et al. Consumer Cash Usage: A Cross-Country Comparison with Payment Diary Survey Data. P. 13. 2014

ESA UNITED NATIONS. World Population Prospects - Population Division. Disponível em: <<https://esa.un.org/unpd/wpp/>>. Acesso em 01 abr. 2018.

FRIEDMAN, Milton. Money Mischief: Episodes in Monetary History. Nova York: Harcourt Brace Jovanovich. p. 16. 1992

IBGE 2017. Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal. Disponível em: <<https://www.ibge.gov.br/estatisticas-novoportal/sociais/trabalho/17270-pnad-continua.html?edicao=19937&t=publicacoes>>. Acesso em 09 set. 2018.

NO SECRET ABOUT QR CODE. Disponível em <<https://www.nosecret.pt/tudo-sobre-qr-codes/>>. Acesso em 01 mar. 2018.

PAYPAL DO BRASIL. About us. Disponível em: <<https://www.paypal.com/us/webapps/mpp/about>>. Acesso em 30 mar. 2018.

EBANX International. Disponível em <<https://www.ebanx.com/br>>. Acesso em 22 abr. 2018.

PETROLL, Luiza Betina. Os cartões de pagamento e a demanda por moeda no Brasil, 1.<sup>a</sup> ed., Cnpq, São Paulo, 2005.

REDAÇÃO. Smartphones estão nas mãos de 62% dos brasileiros, diz Google. Disponível em: <<http://www1.folha.uol.com.br/tec/2017/02/1862362-smartphones-estao-nas-maos-de-62-dos-brasileiros-diz-google.shtml>>. Acesso em 12 jul. 2018.

RIBEIRO, Rorigo. A digitalização de pagamentos e o futuro do papel-moeda. Disponível em <<https://imasters.com.br/devsecops/a-digitalizacao-de-pagamentos-e-o-futuro-do-papel-moeda>>. Acesso em 04 out. 2018.

SCHUN, Scott, & STAVINS, Johanna. How Consumers Pay: Adoption and Use of Payments. pg. 2. 2011

SOUSA, Antonio. Um em cada dois pagamentos na Europa já são 'contactless'. Disponível em <<https://jornaleconomico.sapo.pt/noticias/um-em-cada-dois-pagamentos-na-europa-ja-sao-contactless-355798>>. Acesso em 11 fev. 2018.

THE HISTORY OF QR CODE. Disponível em <<http://www.qrcode.com/en/history/>>. Acesso em 12 abr. 2018.

## APÊNDICES

### APÊNDICE A: VISÃO INICIAL DO PROJETO

A tecnologia vem avançando e a vida real vem se adaptando para esta nova realidade. O que era inimaginável há anos atrás hoje, em muitos casos, é até mesmo obsoleto na tecnologia. Nesse contexto, uma área que vem sendo explorada é a dos pagamentos digitais, a fim de facilitar a concretização de transações por meios eletrônicos.

Objetiva-se construir uma plataforma de transferências financeiras que possa ser utilizada através do smartphone, fazendo uso de *QR codes* e links de pagamento. Um usuário deve poder gerar um código de pagamento em seu celular para que seja escaneado por outro usuário através de seu celular, concretizando a transferência financeira entre estas contas.

Para carregar a sua conta, o usuário poderá gerar, através do aplicativo, um boleto bancário. Para sacar valores, o usuário poderá agendar transferências bancárias para sua conta cadastrada.

## APÊNDICE B: DIAGRAMA DE CASOS DE USO NEGOCIAIS

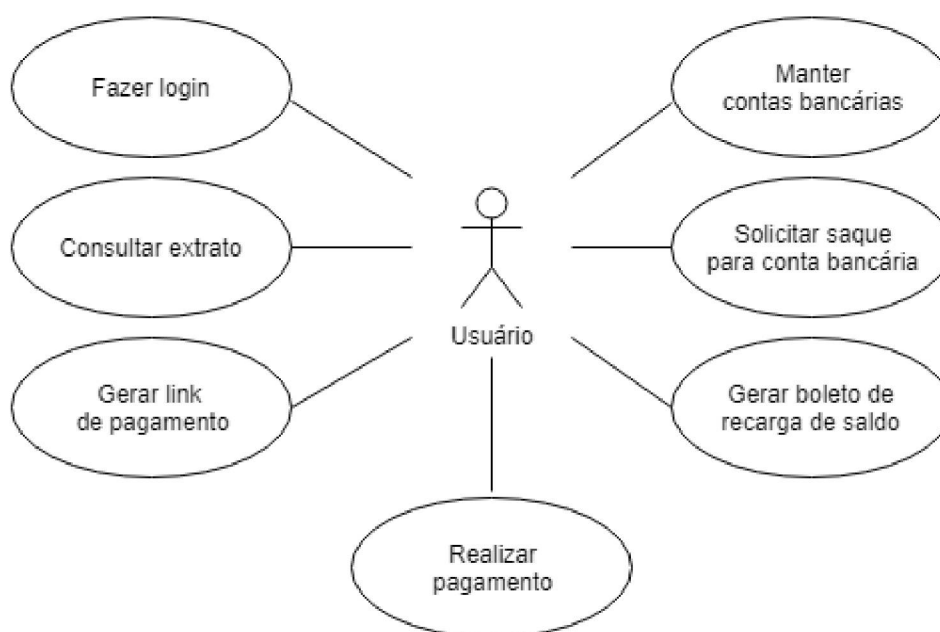


Figura 20: Diagrama de casos de uso comerciais

**UC01 - Consultar extrato:** o usuário pode selecionar um intervalo de até 60 dias para consultar as transações realizadas no período selecionado.

**UC02 - Gerar link de pagamento:** o usuário poderá criar um link de pagamento, que resulta em um *QR code* único, um PIN (código número) e um link de pagamento.

**UC03 - Realizar pagamento:** o usuário poderá escanear um QR code gerado a partir de outra conta, inserir um código PIN ou inserir um link de pagamento.

**UC04 - Manter contas bancárias:** o usuário pode controlar contas bancárias para solicitar saques a qualquer momento.

**UC05 - Solicitar saques para a conta bancária:** o usuário poderá informar o valor do saque desejado. O sistema de apresentar a tarifa administrativa e assim que o usuário confirmar, deve agendar uma transferência bancária para a conta informada.



**UC06 - Gerar boleto de recarga de saldo:** o usuário poderá informar o valor da recarga desejado. O sistema de apresentar a tarifa administrativa e assim que o usuário confirmar, deve gerar um boleto referente aos dados informados.

## APÊNDICE C: GLOSSÁRIO DO SISTEMA

API: Interface de aplicação de programa, do inglês *Application Program Interface*

TARIFA: valor cobrado por uma operação realizada através do aplicativo

PIN: Código numérico único gerado de forma randômica

QR CODE: Código visual de fácil compartilhamento e leitura através de câmeras de smartphones.

## **APÊNDICE D: REGRAS DE NEGÓCIO**

### **UC01 – Consultar extrato**

1. RN1: O período não pode ser superior a 90 dias
2. RN2: Para transações de débito de saldo, o valor deve ser apresentado na cor vermelha. Para as transações de crédito de saldo, o valor deve ser apresentado na cor verde.

### **UC02 - Gerar link de pagamento**

1. RN1: O valor deve ser numérico superior a R\$ 0,00

### **UC03 - Realizar pagamento**

1. RN1: O sistema também deve enviar um e-mail com o comprovante de pagamento

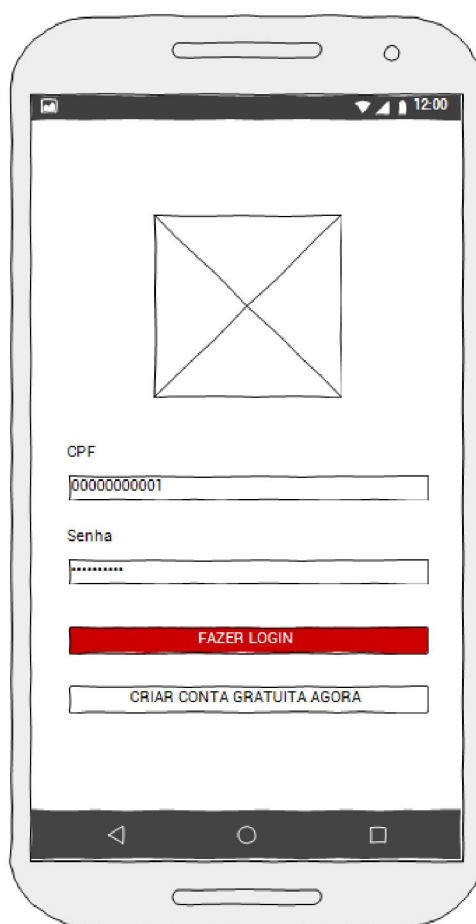
### **UC05 - Solicitar saques para conta bancária**

1. RN1: O valor deve ser numérico deve ser superior a R\$ 0,00
2. RN2: O sistema deve apresentar o código e nome do banco, o número da agência bancária e o número da conta bancária cadastrada do usuário.
3. RN3: O sistema deve enviar um comprovante de saque para o email cadastrado do usuário

### **UC06 - Gerar boleto de recarga de saldo**

1. RN1: O valor deve ser numérico superior a R\$ 0,00
2. RN2: O sistema deve enviar por email o boleto

## APÊNDICE E: PROTÓTIPO DAS PRINCIPAIS INTERFACES



*Figura 21: Protótipo Login*



Figura 22: Protótipo Dashboard

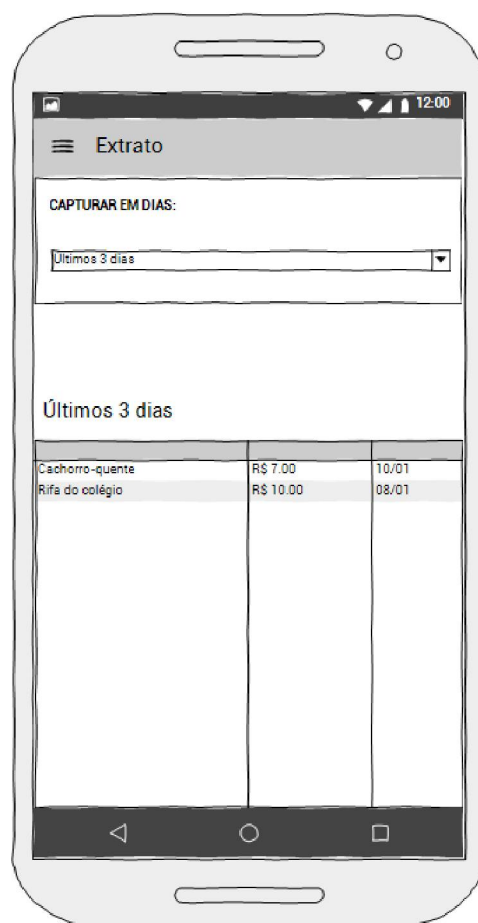


Figura 23: Protótipo da Tela de Extrato

Gerar link de pagamento

Valor total  
R\$ 25,00

Referência  
Casa Praia

Limite de recebimento  
4

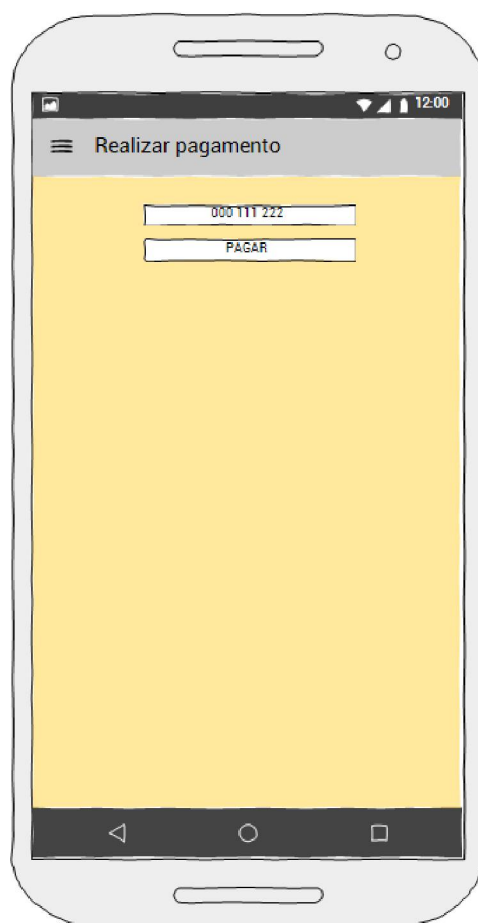
GERAR LINK

Figura 24: Protótipo Gerar link de pagamento



Figura 25: Protótipo Código QR





*Figura 26: Protótipo Realizar pagamento*

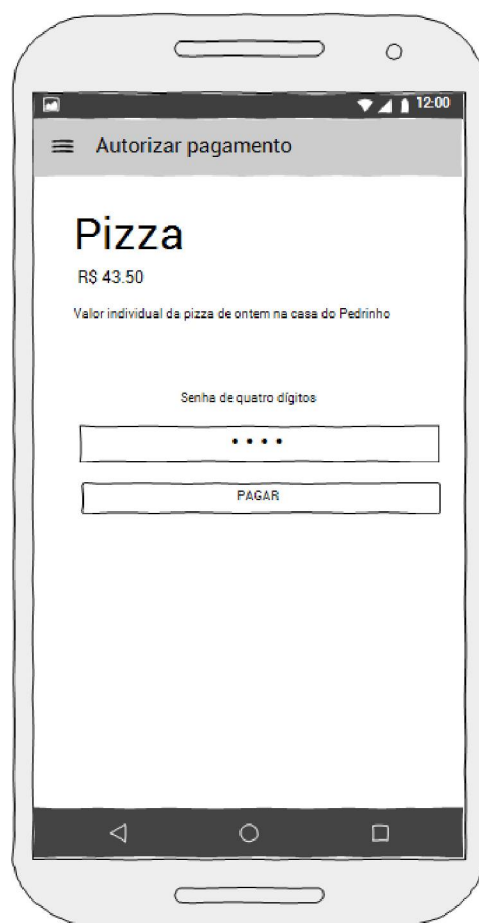


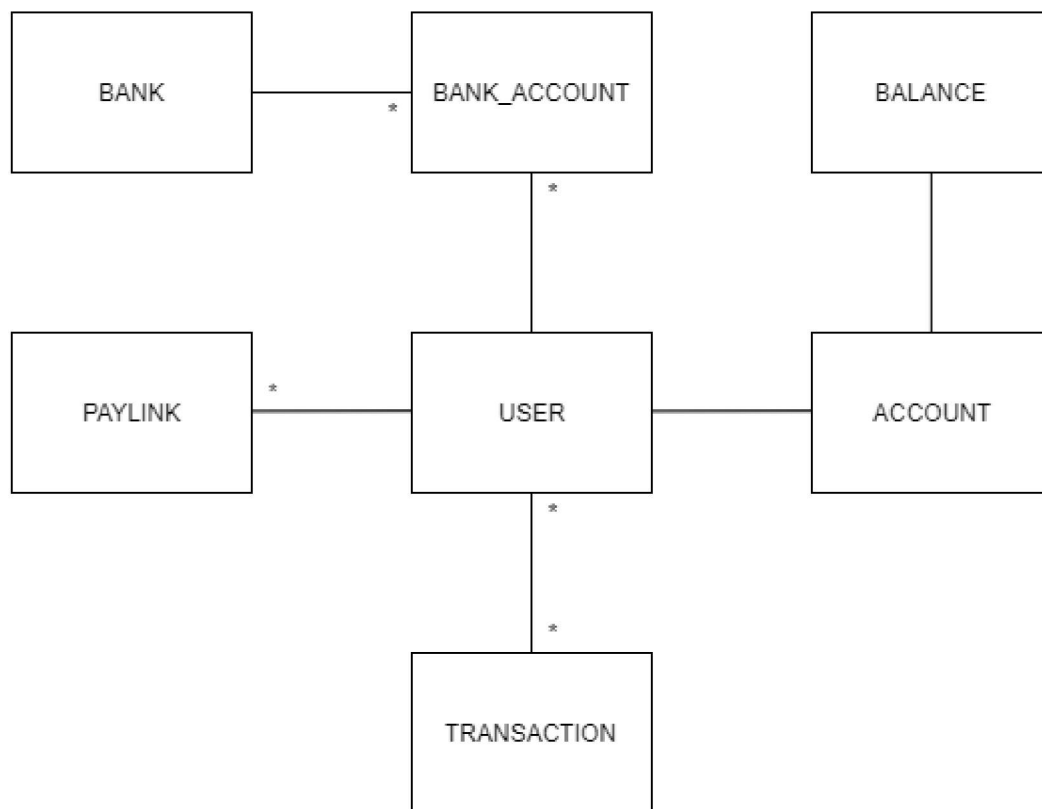
Figura 27: Protótipo Autorizar pagamento

A hand-drawn sketch of a mobile phone displaying a registration form. The phone has a grey border and a black status bar at the top showing signal, battery, and the time 12:00. The form is centered on the screen and consists of the following elements:

- Nome**: A text input field containing "John Doe".
- E-mail**: A text input field containing "johndoe@example.com".
- CPF**: A text input field containing "12312312387".
- Senha**: A text input field with seven dots for password masking.
- Senha de quatro dígitos**: A text input field with four dots for a 4-digit PIN.
- Button**: A rectangular button with the text "CRIAR CONTA GRATUITA AGORA".

At the bottom of the phone screen is a black navigation bar with three white icons: a triangle (back), a circle (home), and a square (recent apps).

Figura 28: Protótipo Cadastrar conta

**APÊNDICE F: DIAGRAMA DE CLASSES DE OBJETOS DE NEGÓCIO***Figura 29: Diagrama de Classes Nível I*

## APÊNDICE G: DIAGRAMA DE CLASSES COM ATRIBUTOS

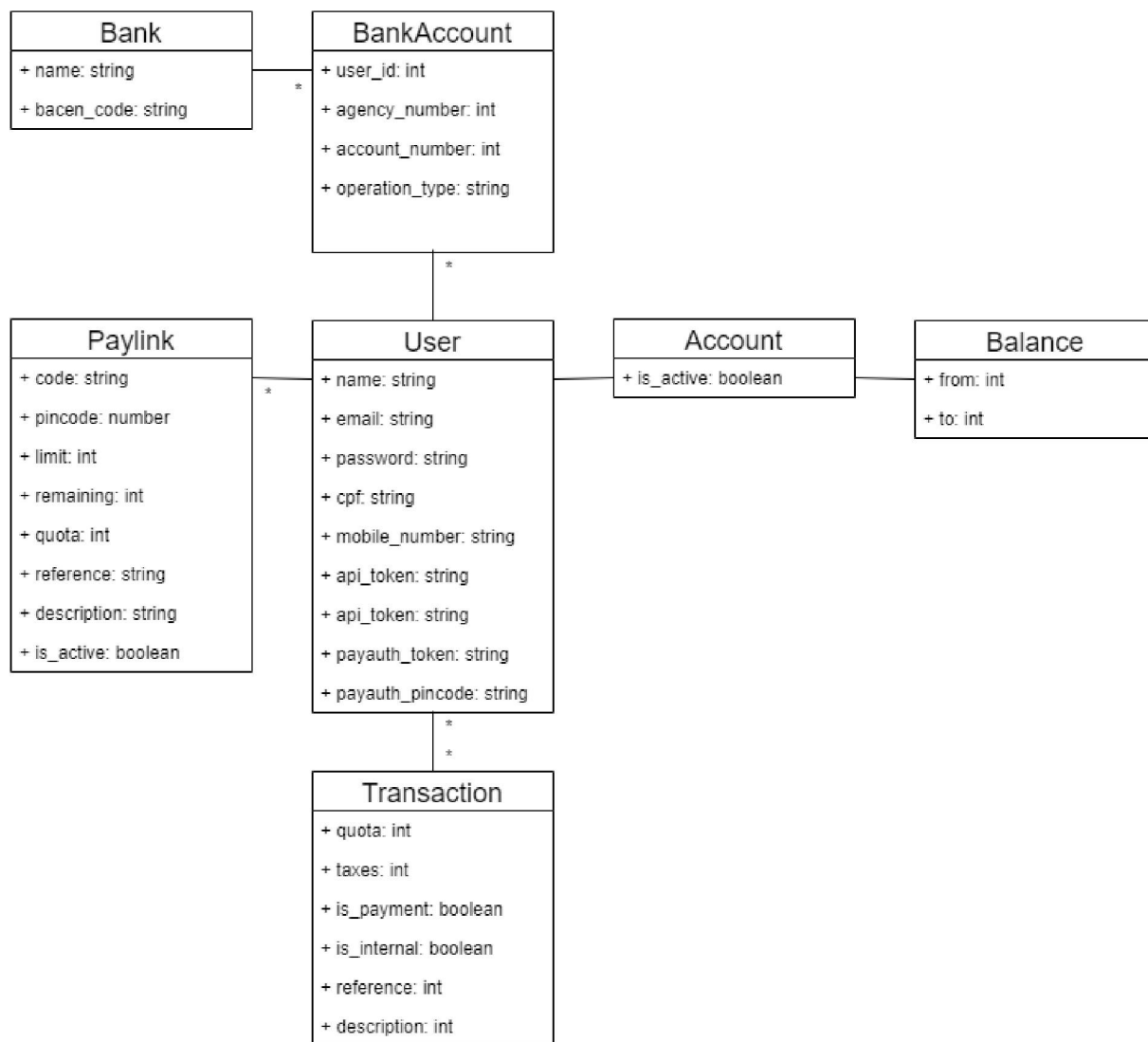


Figura 30: Diagrama de Classes Nível II

## APÊNDICE H: DIAGRAMA DE CLASSES COMPLETO

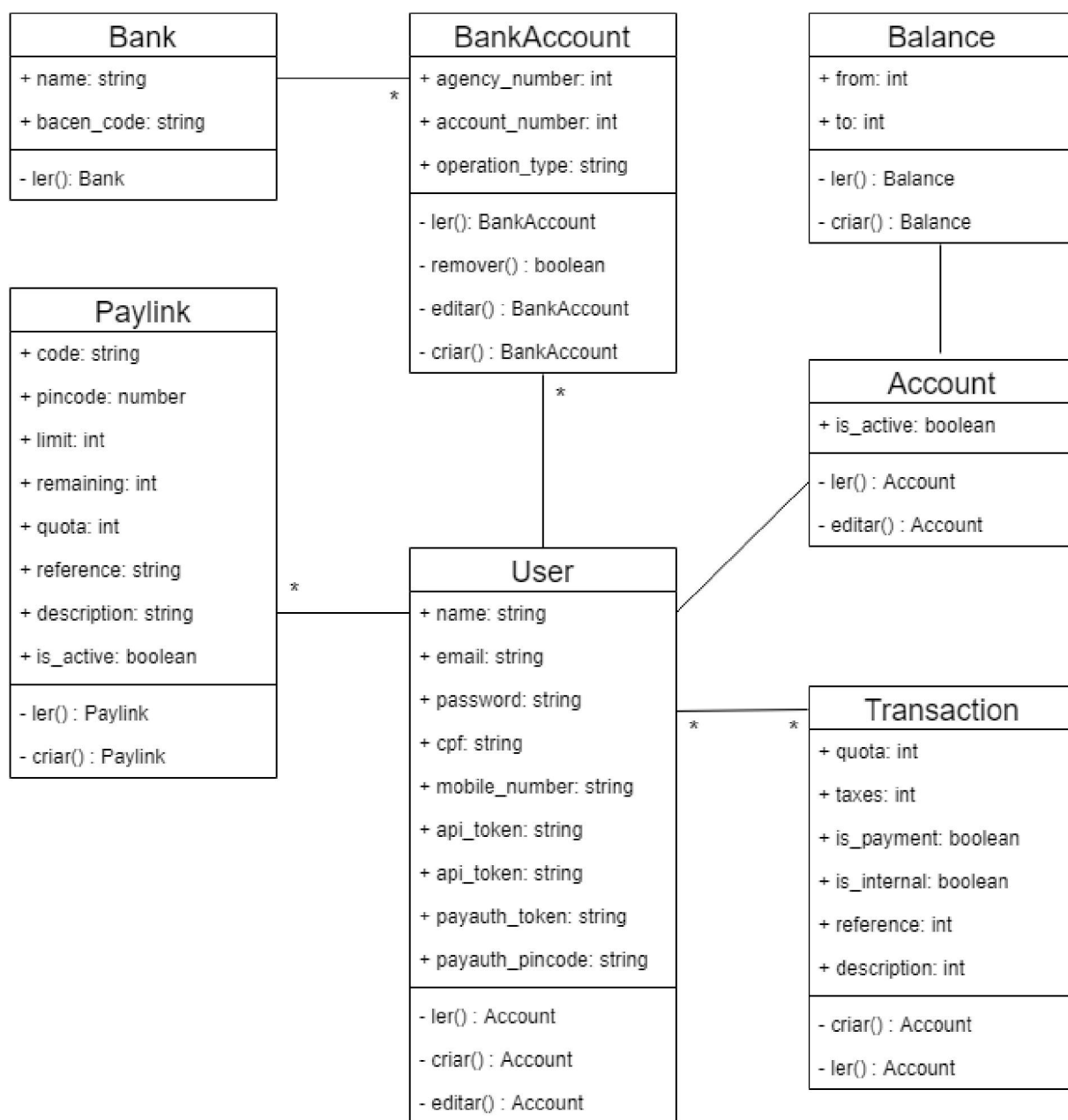


Figura 31: Diagrama de Classes Nível III

## APÊNDICE I: ESPECIFICAÇÃO DOS CASOS DE USO

### UC01 – Consultar extrato

#### Descrição

Esse caso de uso descreve o processo de consulta ao extrato de transações efetuadas pelo usuário

#### Data View

← Extrato de transações		
Capturar em dias:		
Últimos 15 dias		
Últimas transações (Últimos 15 dias)		
Recarga de Conta	R\$ 150.00	11/10/2018
<small>Boleto Bancário</small>		
Link gerado manualmente por outro user 3	R\$ 30.00	11/10/2018
<small>Descrição manual também por outro user 3</small>		
Link gerado manualmente por outro user 3	R\$ 30.00	11/10/2018
<small>Descrição manual também por outro user 3</small>		
Link gerado manualmente por outro user 3	R\$ 30.00	11/10/2018
<small>Recebimento via PagFit</small>		
Link gerado manualmente por outro user 3	R\$ 30.00	15/10/2018
<small>Descrição manual também por outro user 3</small>		

#### Pré-condições

Estar logado no aplicativo como usuário regular

#### Pós-condições

O aplicativo deve apresentar um extrato consolidado de todas as transações ocorridas no período selecionado

**Ator primário:** Usuário

#### Fluxo de eventos principal

1. O sistema exibe a tela (DV1)
2. O usuário seleciona um intervalo de consulta (RN1)

3. O sistema apresenta todas as transações ocorridas no período selecionado (RN2).
4. O caso de uso é encerrado

### **Fluxo de eventos alternativo**

O caso de uso não possui fluxos alternativos

### **Regras de negócio**

3. RN1: O período não pode ser superior a 90 dias
4. RN2: Para transações de débito de saldo, o valor deve ser apresentado na cor vermelha. Para as transações de crédito de saldo, o valor deve ser apresentado na cor verde.

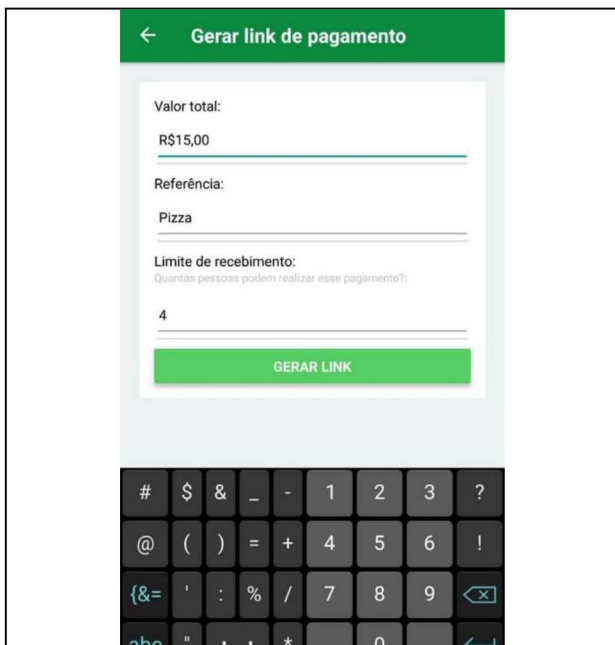
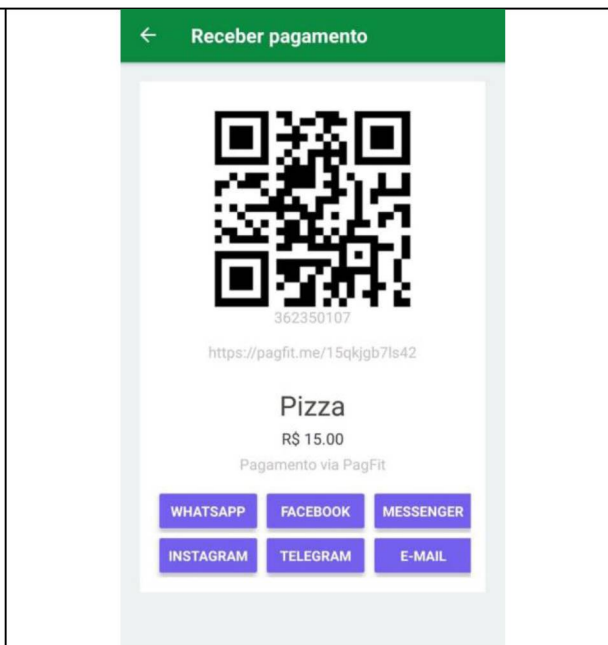


## UC02 - Gerar link de pagamento

### Descrição

Esse caso de uso descreve o processo de geração de link de transferência de entrada de valores (link de pagamento)

### Data View

	
--	---

### Pré-condições

Estar logado no aplicativo como usuário regular

### Pós-condições

O aplicativo deve ter gerado um código QR de pagamento cuja referência aponta para um link de pagamento no sistema,

**Ator primário:** Usuário

### Fluxo de eventos principal

1. O sistema exibe a tela (DV1)
2. O usuário insere um valor (RN1)
3. O usuário preenche os demais campos da tela
4. O usuário clica em "Gerar link"

5. O sistema valida as informações e direciona para a tela de confirmação (DV2)
6. O sistema apresenta um código QR e sugestões de compartilhamento em redes sociais do dispositivo.
7. O caso de uso é encerrado

**Fluxo de eventos alternativo**

O caso de uso não possui fluxos alternativos

**Regras de negócio**

1. RN1: O valor deve ser numérico superior a R\$ 0,00

## UC03 - Realizar pagamento

### Descrição

Esse caso de uso descreve o processo de concretização de transferência de saída de valores (pagamento)

### Data View

The image displays three sequential screens of a mobile application for making a payment:

- Realizar pagamento:** Features a green header with a back arrow and the title. Below is a white input field containing the number '100114100'. A purple button labeled 'PAGAR' is positioned below the input field.
- Autorizar pagamento:** Features a green header with a back arrow and the title. The main content area is white and displays: 'Link gerado manualmente por outro user 3', 'R\$ 30.00', 'Descrição manual também por outro user 3', and 'Senha de quatro dígitos:' followed by four dots. A green button labeled 'PAGAR' is at the bottom.
- Confirmação de pagamento:** Features a green header with a back arrow and the title. The main content area is white and displays: 'Pagamento concluído!', a list of details (Nome: Test User 1, Cidade/UF: Curitiba/PR, CPF: 00000000001, Valor total: R\$ 30.00, Referência: Link gerado manualmente por outro user 3, Descrição: Descrição manual também por outro user 3), and a note stating the receipt was sent to the user's email. A red button labeled 'VISUALIZAR PDF' is at the bottom.

### Pré-condições

Estar logado no aplicativo como usuário regular e possuir saldo positivo em conta

### Pós-condições

O aplicativo deve ter debitado o valor do pagamento da conta do usuário, creditado o valor na conta-destino, registrar a transferência no histórico de transações e enviar dois emails com o comprovante, sendo um para o usuário outro para o destinatário do valor.

**Ator primário:** Usuário

**Fluxo de eventos principal**

1. O sistema exibe a tela (DV1)
2. O usuário aponta o dispositivo para um código QR (A1)
3. O sistema lê as informações e direciona para a tela de confirmação (DV2) (A2)
4. O usuário confere os dados, digita uma senha de quatro dígitos e pressiona em “Pagar” (A3)
5. O sistema debita o valor da conta do usuário e apresenta comprovante de pagamento (RN1)
6. O caso de uso é encerrado

**Fluxo de eventos alternativo****A1. Usuário insere link de pagamento ao invés de código QR**

1. O sistema consulta as informações do link de pagamento e redireciona para a tela de confirmação (A2)
2. O caso de uso é retornado ao passo 3 do fluxo principal

**A2. Usuário sem saldo suficiente**

1. O sistema exibe a mensagem “Saldo insuficiente para realizar este pagamento”
2. O caso de uso é encerrado.

**A3. Senha de quatro dígitos inválida**

1. O sistema exibe a mensagem “Senha inválida”
2. O caso de uso é retornado ao passo 3 do fluxo principal

**Regras de negócio**

1. RN1: O sistema também deve enviar um e-mail com o comprovante de pagamento

## UC04 - Manter contas bancárias

### Descrição

Esse caso de uso descreve o processo de manutenção de contas bancárias do usuário

### Data View

The image displays three sequential mobile application screens for managing bank accounts. The first screen, titled 'Contas bancárias', shows a list of accounts with one entry: '237 - Banco Bradesco S.A. Ag 588 - CC 545066'. Below the list is a button labeled '(+) Nova conta bancária'. The second screen, titled 'Gerenciar conta bancária', shows a form for editing an account. It contains three input fields: 'Número da agência:' with the value '1223', 'Número da conta:' with the value '3344556', and 'Código BACEN do banco:' with the value '104'. A green 'SALVAR' button is at the bottom. The third screen, also titled 'Gerenciar conta bancária', shows the same form with the 'SALVAR' button and an additional red 'EXCLUIR' button at the bottom.

### Pré-condições

Estar logado no aplicativo como usuário regular

### Pós-condições

O aplicativo deve ter realizado a operação de inserção, atualização ou remoção de uma conta bancária do usuário

**Ator primário:** Usuário

### Fluxo de eventos principal

1. O sistema exibe a tela (DV1)
2. O usuário seleciona uma conta bancária (A1)
3. O sistema apresenta um formulário de edição (DV3)
4. O usuário preenche os campos da tela

5. O usuário clica em “Salvar” (A2)
6. O sistema valida as informações e direciona para a tela de listagem (DV1)
7. O caso de uso é encerrado

#### **Fluxo de eventos alternativo**

##### **A1. Usuário clica em Nova conta bancária**

1. O sistema apresenta um formulário de cadastro (DV2)
2. O caso de uso retorna ao Fluxo principal no passo 4

##### **A2. Usuário clicou em Excluir**

1. O sistema remove a conta bancária
2. O caso de uso retorna ao Fluxo principal no passo 6

## UC05 - Solicitar saques para conta bancária

### Descrição

Esse caso de uso descreve o processo de solicitação de operação de depósito bancário (saque)

### Data View

--	--

### Pré-condições

Estar logado no aplicativo como usuário regular e possui saldo em conta para efetuar o depósito.

### Pós-condições

O aplicativo deve ter gerado um registro interno de depósito para conta bancária, bem como debitar o saldo do usuário. Também deve acrescentar a transação no histórico de transação do cliente

**Ator primário:** Usuário

### Fluxo de eventos principal

1. O sistema exibe a tela (DV1)
2. O usuário insere um valor (RN1)

3. O usuário seleciona uma das contas bancárias exibidas na tela (RN2)
4. O usuário clica em “Solicitar saque”
5. O sistema valida as informações e direciona para a tela de confirmação (DV2) (A1) e apresenta novamente os dados bancários (RN2) (RN3)
6. O caso de uso é encerrado

### **Fluxo de eventos alternativo**

#### **A1. Usuário sem saldo suficiente**

1. O sistema exibe a mensagem “Saldo insuficiente para realizar este saque”
2. O caso de uso é encerrado.

### **Regras de negócio**

4. RN1: O valor deve ser numérico deve ser superior a R\$ 0,00
5. RN2: O sistema deve apresentar o código e nome do banco, o número da agência bancária e o número da conta bancária cadastrada do usuário.
6. RN3: O sistema deve enviar um comprovante de saque para o email cadastrado do usuário



## UC06 - Gerar boleto de recarga de saldo

### Descrição

Esse caso de uso descreve o processo de geração de boleto de transferência de entrada de valores direto na conta (recarga de conta)

### Data View

--	--

### Pré-condições

Estar logado no aplicativo como usuário regular

### Pós-condições

O aplicativo deve ter gerado um boleto bancário para recarga de saldo na conta

**Ator primário:** Usuário

### Fluxo de eventos principal

1. O sistema exibe a tela (DV1)
2. O usuário insere um valor (RN1)
3. O usuário clica em “Gerar boleto”
4. O sistema valida as informações e direciona para a tela de confirmação (DV2)

5. O sistema apresenta uma linha digitável do boleto para facilitar pagamentos em aplicativos relacionados (RN2).
6. O caso de uso é encerrado

**Fluxo de eventos alternativo**

O caso de uso não possui fluxos alternativos

**Regras de negócio**

1. RN1: O valor deve ser numérico superior a R\$ 0,00
2. RN2: O sistema deve enviar por email o boleto.

## APÊNDICE J: DIAGRAMAS DE SEQUÊNCIA

## UC01 – Consultar extrato

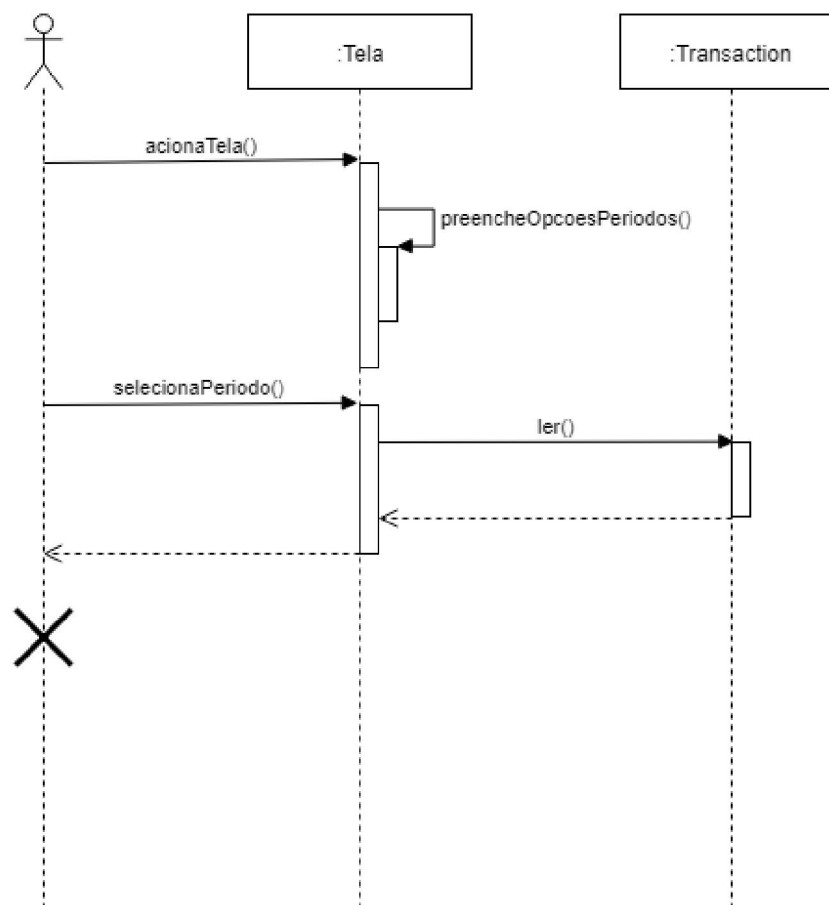
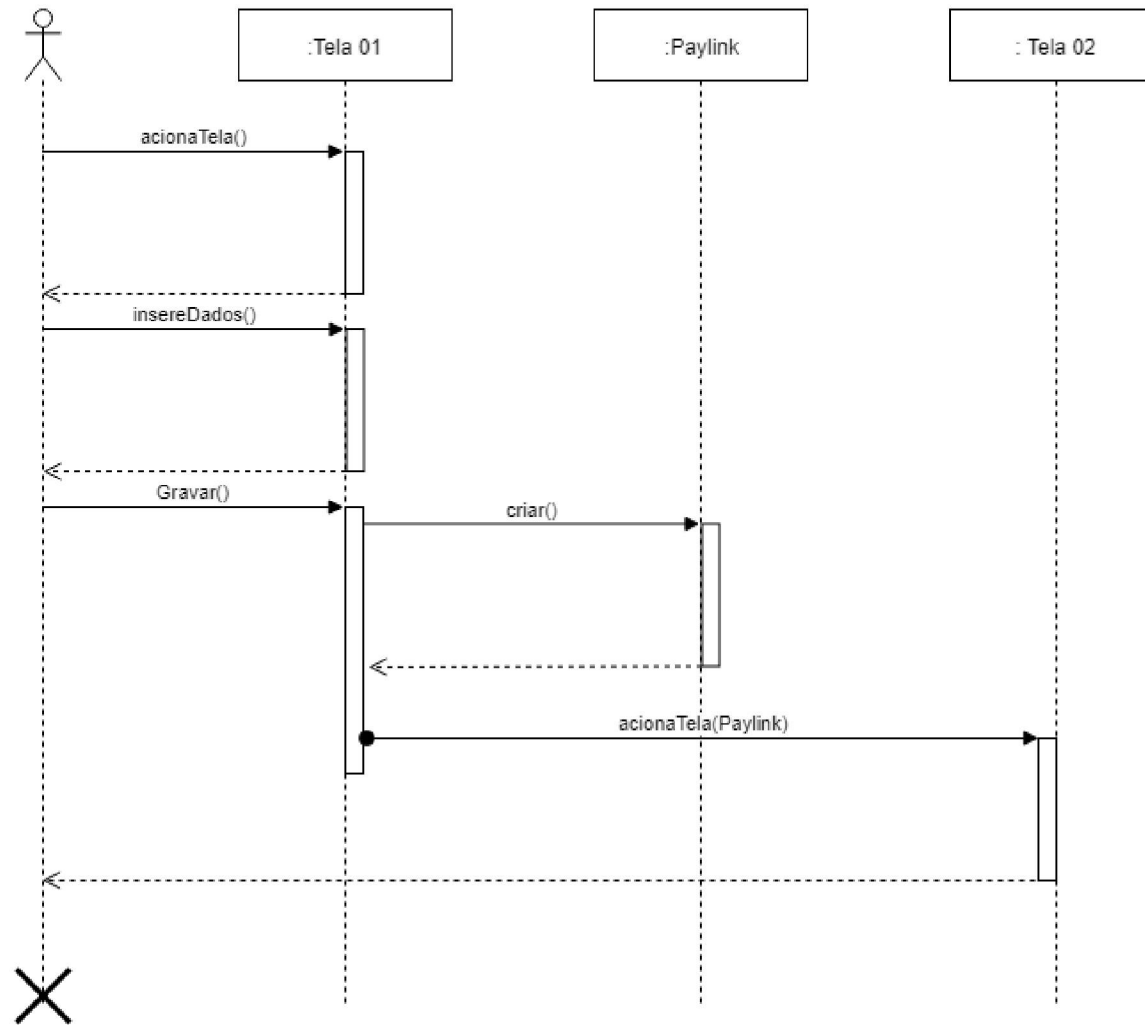
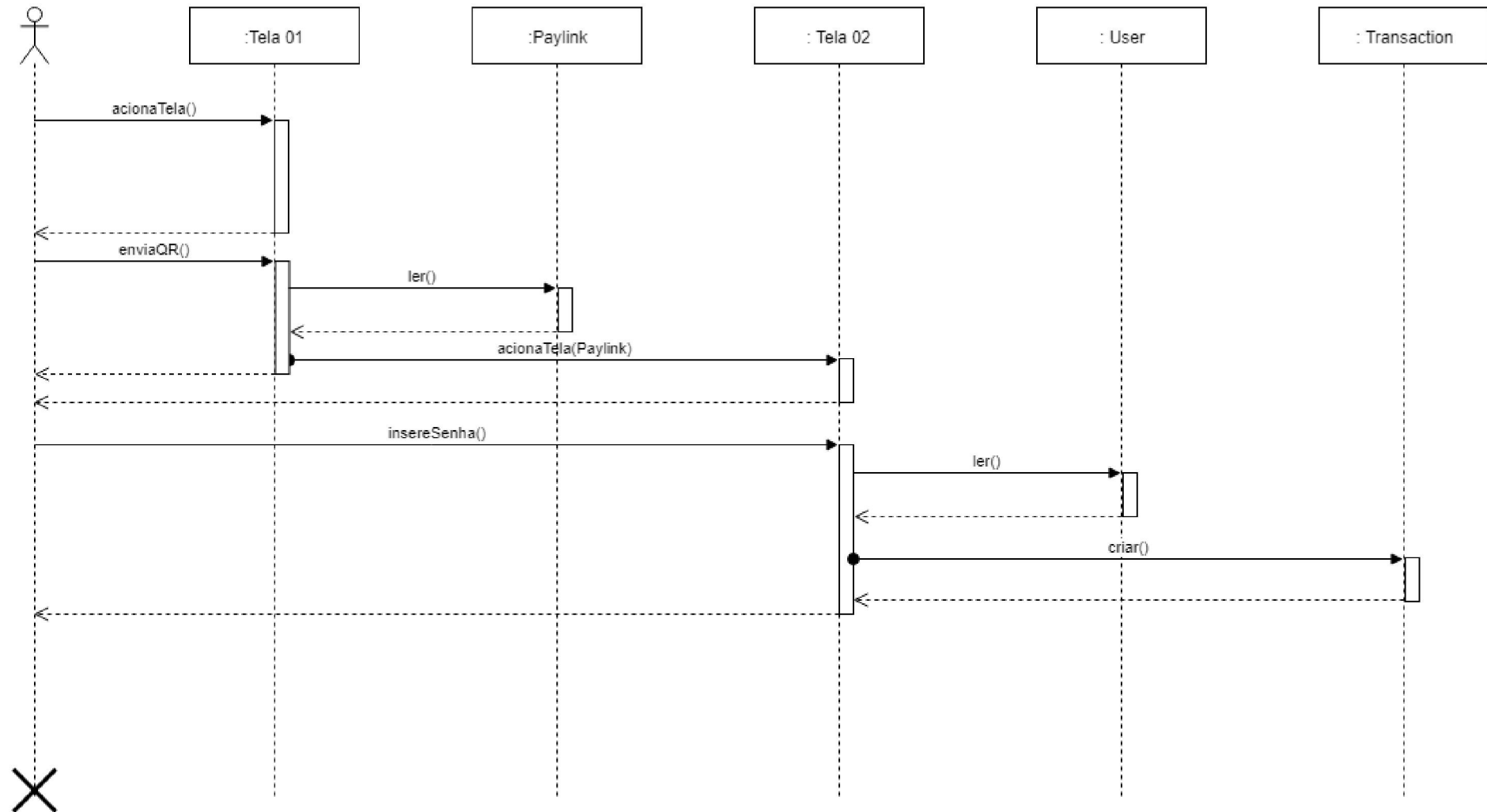


Figura 32: Diagrama de Sequência UC01

**UC02 - Gerar link de pagamento***Figura 33: Diagrama de Sequência UC02*

**UC03 - Realizar pagamento***Figura 34: Diagrama de Sequência UC03*

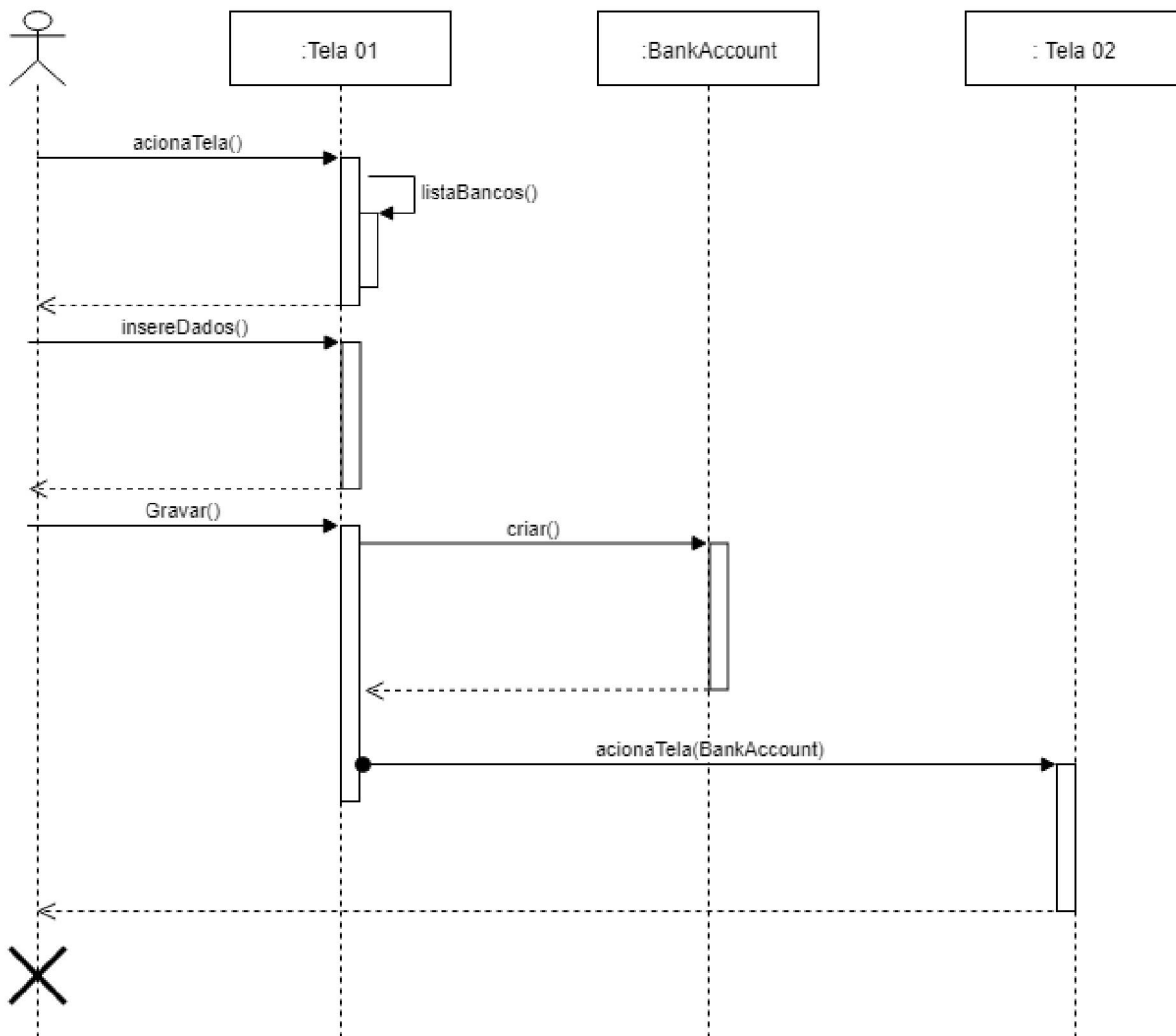
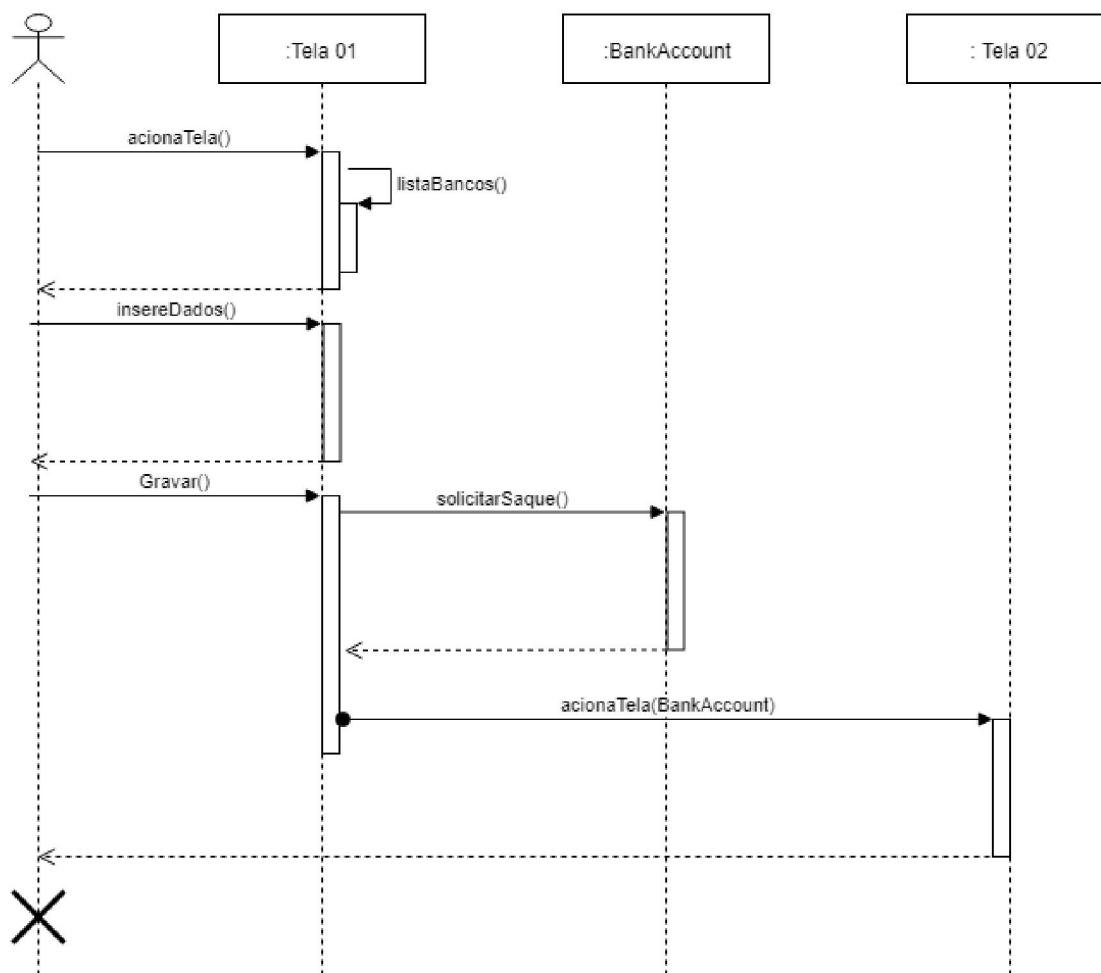
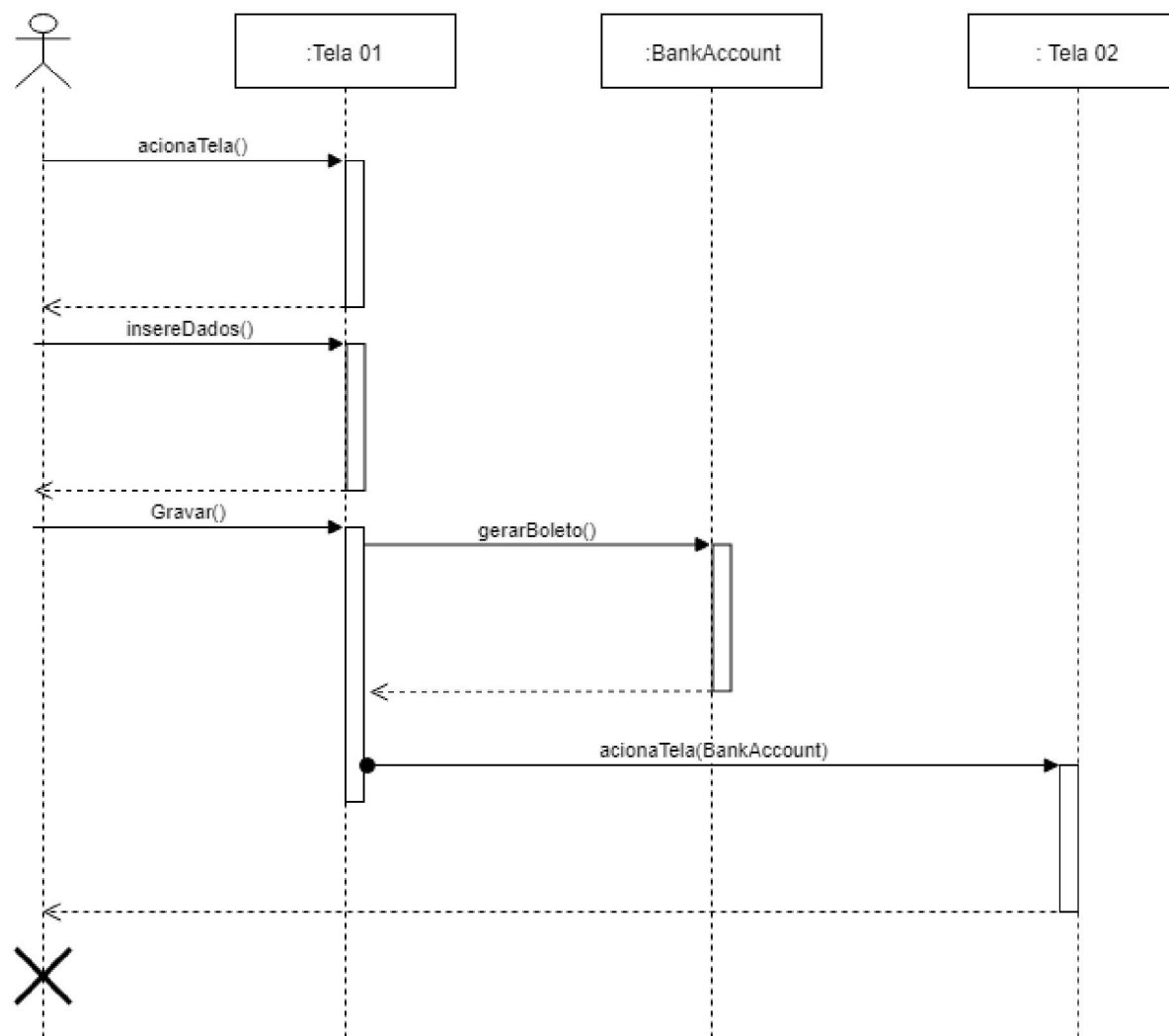
**UC04 – Manter contas bancárias**

Figura 35: Diagrama de Sequência UC04 – Listar contas

**UC05 - Solicitar saques para conta bancária***Figura 366: Diagrama de Sequência UC05*

**UC06 - Gerar boleto de recarga de saldo***Figura 377: Diagrama de Sequência UC06*



## APÊNDICE K: MODELO FÍSICO DE DADOS

Como a aplicação não consome o banco de dados diretamente, a fonte dos dados é a API oficial da plataforma. A mesma oferece uma estrutura de objetos (modelo físico), conforme descrito na sequência:

### User

```
{
  "name": [string],
  "email": [string],
  "birthdate": [string:8],
  "city": [string],
  "state": [string:2],
  "cpf": [number:11],
  "mobile_number": [number:11],
  "api_token": "Bearer [string:bcrypt]",
  "account": Account<>,
  "bank_account": [BankAccount<>]
}
```

### Account

```
{
  "balance": [int],
  "is_active": [boolean]
}
```

### BankAccount

```
{
  "id": [int],
  "agency_number": [number],
  "account_number": [number],
  "bank": Bank<>
}
```

### Bank

```
{
  "id": [int],
```

```

    "name": [string],
    "bacen_code": [number]
}

```

## Transaction

```

{
    "id": [int],
    "origin_account_id": [int],
    "target_account_id": [int],
    "quota": [int],
    "is_payment": [boolean],
    "is_internal": [boolean],
    "reference": [string],
    "description": [string],
    "created_at": [datetime],
    "updated_at": [datetime]
}

```

## Paylink

```

{
    "id": [int],
    "code": [string],
    "pincode": [int],
    "quota": [int],
    "reference": [string],
    "description": [string],
    "limit": [int],
    "remaining": [int],
    "is_active": [boolean],
    "user": User<>
}

```

## Withdrawal

```

{
    "identifier": [int],
    "quota": [int],
    "transfer_date": [number:8],
    "bank_account": BankAccount<>
}

```

**Billet**

```
{
  "identifier": [number],
  "quota": [int],
  "barcode": [string],
  "expire_date": [number:8]
}
```

**Rotas**

- Login - POST /api/login
- Register - POST /api/register
- Get User Info - GET /api/user
- Edit User - POST /api/user
- Get Basic User Info - GET /api/user/basic?cpf=:CPF
- Get History - GET /api/user/history?from=:SINCE
- Create Bank Account - POST /api/bank\_account
- List User Bank Accounts - GET /api/bank\_account
- Get User Bank Account by Id - GET /api/bank\_account/:ID
- Update Bank Account by Id - POST /api/bank\_account/:ID
- Delete Bank Account by Id - DELETE /api/bank\_account/:ID
- Create Paylink – POST /api/paylink
- List User Paylinks – GET /api/paylink
- Get Paylink by [Params] – GET /api/paylink/:CODE
- Update Paylink by Id - POST /api/paylink/:ID
- Delete Paylink by Id – DELETE /api/paylink/:ID
- Get Payauth Token – POST /api/ops/auth
- Pay – POST /api/ops/pay
- Create Billet – POST /api/ops/billet
- Schedule Withdrawal – POST /api/ops/withdrawal

## APÊNDICE L: PLANO DE TESTES

### Plano de Exemplo - TC01

**Caso de Uso:** UC03 Realizar pagamento

**Pré-condições:** Estar logado no aplicativo como usuário regular e possuir saldo positivo em conta

Os testes serão efetuados em três baterias.

- Tipo de teste: Unitário
- Ferramenta de teste: Jest
- Análise dos dados: Através de logs e CLI das ferramentas

### BATERIA 01: TODOS OS CASOS - NÃO ANALÍTICO

**PROPOSTA:** Assegurar a qualidade dos testes, a fim de poder realizar a análise crítica das demais baterias e assegurar a completude do sistema. Executar os casos de uso e garantir a execução correta dos mesmos.

### BATERIA 02: CASOS POSITIVOS - ANÁLISE DE RESULTADOS COERENTES

**PROPOSTA:** Analisar criticamente os resultados obtidos para assegurar a qualidade do sistema. Executar casos de teste 01, 02, 03 e 06

### CASO TC01.01

O sistema deve assegurar que possui permissão do usuário para acessar a câmera do dispositivo em questão. Uma vez que essa informação é fornecida pelo navegador do usuário e independente do servidor, basta checar se há a permissão em questão.

### CASO TC01.02

O sistema deve realizar o processamento do código Quick Read, a fim de localizar o atributo identificador encapsulado. Por se tratar de um QR válido, o processamento será bem-sucedido e por consequência irá permitir ao usuário realizar o pagamento.

**CASO TC01.03**

O sistema deve realizar o parseamento do link, a fim de localizar o atributo identificador do link. Por se tratar de um link válido, o parseamento será bem-sucedido e por consequência irá permitir ao usuário realizar o pagamento.

**CASO TC01.06**

O sistema deve oferecer ao usuário a liberdade de cancelar a ação do pagamento. Por isso, o caso de teste assegura que o usuário tem acesso a uma tela de confirmação de pagamento.

**CASO TC01.07**

O sistema deve oferecer ao usuário a liberdade de cancelar a ação do pagamento. Por isso, o caso de teste assegura que o usuário CANCELE a operação.

**CASO TC01.08**

O sistema deve oferecer ao usuário a liberdade de cancelar a ação do pagamento. Por isso, o caso de teste assegura que o usuário conclua a operação.

**BATERIA 03: CASOS NEGATIVOS - ANÁLISE DE RESULTADOS INCOERENTES**

**PROPOSTA:** Analisar criticamente os resultados obtidos para assegurar a eficiência do sistema. Executar casos de teste 04 e 05 utilizando LINKS DE PAGAMENTO

**CASO TC01.04**

O sistema deve realizar o parseamento do link, a fim de localizar o atributo identificador do link. Por se tratar de um link inválido, o parseamento falhará e por consequência irá prevenir o usuário de realizar o pagamento.

**CASO TC01.05**

O sistema deve assegurar a transação de forma segura e válida, portanto deve rejeitar transações cujo emitente não possua saldo em conta.

## APÊNDICE M: CASOS DE TESTE

### Plano de Exemplo - TC01

<b>Caso de Uso</b>	UC04 Realizar pagamento
<b>Pré-condições</b>	Estar logado no aplicativo como usuário regular e possuir saldo positivo em conta

Nº	Pré-condições	Entrada	Ação	Resultado Esperado
01	O usuário deve conceder permissão de acesso à câmera			A tela deve exibir a imagem da câmera em fullscreen
02		A câmera deve ter lido um código QR válido	Apontar para o código	Redirecionar a tela para o pagamento correspondente ao código
03		O usuário deve inserir um código PIN válido  Ex; 123123123	Pressionar "Pagar"	Redirecionar a tela para o pagamento correspondente ao código
04		O usuário deve inserir um link inválido  <a href="https://invalidlink.com">https://invalidlink.com</a>	Pressionar "Pagar"	A tela deve retornar um aviso "Link inválido ou incompatível"
05	Passo 02 ou 03 realizado com link cujo valor seja superior ao saldo do usuário de teste			A tela deve retornar um aviso "Saldo insuficiente para realizar este pagamento"
06	Passo 02 ou 03 realizado com link cujo valor seja igual ou menor ao saldo do usuário de teste			A tela deve apresentar as informações de pré-pagamento
07	Passo 06		Pressionar "Voltar" no menu do DISPOSITIVO	A tela deve redirecionar a tela inicial, cancelando a intenção de pagamento
08	Passo 06		Pressionar "Confirmar"	O sistema deve redirecionar para a tela de confirmação de pagamento

## APÊNDICE N: LOG DE TESTES

O autor realizou a bateria de testes conforme planejado no Apêndice L, conforme especificações do Apêndice M. Os resultados foram satisfatórios, conforme o quadro a seguir:

Nº	Pré-condições	Entrada	Resultado Esperado	Resultado Obtido
01	O usuário deve conceder permissão de acesso à câmera		A tela deve exibir a imagem da câmera em fullscreen	A tela exibiu a imagem da câmera em fullscreen
02		A câmera deve ter lido um código QR válido	Redirecionar a tela para o pagamento correspondente ao código	Redirecionou a tela para o pagamento correspondente ao código
03		O usuário deve inserir um código PIN válido  Ex; 100114100	Redirecionar a tela para o pagamento correspondente ao código	Redirecionou a tela para o pagamento correspondente ao código
04		O usuário deve inserir um PIN inválido  Ex: 0	A tela deve retornar um aviso "Link inválido ou incompatível"	A tela retornou um aviso "Link inválido ou incompatível"
05	Passo 02 ou 03 realizado com link cujo valor seja superior ao saldo do usuário de teste		A tela deve retornar um aviso "Saldo insuficiente para realizar este pagamento"	A tela retornou um aviso "Saldo insuficiente para realizar este pagamento"
06	Passo 02 ou 03 realizado com link cujo valor seja igual ou menor ao saldo do usuário de teste		A tela deve apresentar as informações de pré-pagamento	A tela apresentou as informações de pré-pagamento
07	Passo 06		A tela deve redirecionar a tela inicial, cancelando a intenção de pagamento	A tela redirecionou para a tela inicial, cancelando a intenção de pagamento
08	Passo 06		O sistema deve redirecionar para a tela de confirmação de pagamento	O sistema redirecionou para a tela de confirmação de pagamento

## Resultado dos testes

Os testes foram realizados em ambiente local e de produção e em ambos, o resultado esperado foi obtido com sucesso.

← Realizar pagamento	← Autorizar pagamento	← Confirmação de pagamento
<div>100114100</div> <div>PAGAR</div>	<div>Link gerado manualmente por outro user 3</div> <div>R\$ 30.00</div> <div>Descrição manual também por outro user 3</div> <div>Senha de quatro dígitos: ....</div> <div>PAGAR</div>	<div>Pagamento concluído!</div> <div>- Nome: Test User 1</div> <div>- Cidade/UF: Curitiba/PR</div> <div>- CPF: 00000000001</div> <div>- Valor total: R\$ 30.00</div> <div>- Referência: Link gerado manualmente por outro user 3</div> <div>- Descrição: Descrição manual também por outro user 3</div> <div>Esse comprovante também foi enviado para seu e-mail de cadastro</div> <div>VISUALIZAR PDF</div>



## APÊNDICE O: PLANO DE IMPLANTAÇÃO

O autor desse projeto planeja disponibilizar o aplicativo para uso realizando o procedimento de publicação na Play Store, loja de aplicativos do Google e na App Store, loja de aplicativos da Apple.

Para isso, será realizado o *build* do aplicativo através do comando `yarn eject`, que converte o código JavaScript em Java Android (Android) e Swift (iOS). Após, para cada plataforma em questão será feito o procedimento de compilação do código.